Installation

Step 1

nano deploy_monitoring.sh

Step 2 (copy the script to the the file and save and exit)

```
#!/bin/bash
# Exit immediately if a command exits with a non-zero status.
set -e

echo "--- Starting Full Monitoring Stack Deployment (v7 - Assisted) ---"

# --- 1. Install Dependencies and Incus ---
echo "Updating packages and installing dependencies..."
# We must run as root, so sudo is removed from apt-get
apt-get update
apt-get install -y spice-vdagent spice-webdavd wget btrfs-progs curl

echo "Adding Zabbly repository for Incus..."
wget -O /etc/apt/keyrings/zabbly.asc https://pkgs.zabbly.com/key.asc

# This command needs sudo because it's a sub-shell
sudo sh -c 'cat <<EOF > /etc/apt/sources.list.d/zabbly-incus-stable.sources
Enabled: yes
Types: deb
URIs: https://pkgs.zabbly.com/incus/stable
Suites: $(. /etc/os-release && echo ${VERSION_CODENAME})
Components: main
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/zabbly.asc
EOF'

echo "Installing Incus..."
apt-get update
apt-get install -y incus

# --- 2. MANUAL STEP ---
echo "---"
echo "--- IMPORTANT: MANUAL ACTION REQUIRED ---"
echo ""
echo "The script must now pause."
echo "Please open a NEW terminal window and log in to this same server."
echo ""
echo "    sudo incus admin init"
echo ""
echo "Answer the questions with these exact values:"
echo "  - Clustering: no"
```

```
echo "  - New storage pool: yes"
echo "  - Storage pool name: default"
echo "  - Storage backend: btrfs"
echo "  - Create new BTRFS pool: yes"
echo "  - Use existing block device: no"
echo "  - Size of loop device: 20"
echo "  - New local network bridge: yes"
echo "  - Bridge name: incusbr0"
echo "  - IPv4: auto"
echo "  - IPv6: auto"
echo "  - Available over network: no"
echo "  - Update stale images: yes"
echo "  - Print YAML preseed: no"
echo ""
read -p "After you have completed 'incus admin init' in the OTHER terminal, press [ENTER]
in THIS window to continue..."
echo ""
echo "--- Manual step complete. Resuming automation... ---"

# --- 3. Create and Start Containers ---
echo "Creating Incus containers..."
incus create images:ubuntu/noble/cloud switch
incus create images:ubuntu/noble/cloud cnt2
incus create images:ubuntu/noble/cloud SNMPExporter
incus create images:ubuntu/noble/cloud alertmanager

echo "Starting containers..."
incus start switch SNMPExporter alertmanager cnt2

# Wait a few seconds for containers to get network access
echo "Waiting 10s for containers to boot and get DNS..."
sleep 10

# --- 4. Install Prometheus (on Host) ---
echo "Installing Prometheus on the host..."
apt-get install -y wget tar
wget
https://github.com/prometheus/prometheus/releases/download/v2.54.1/prometheus-2.54.1.li
nux-amd64.tar.gz
tar xvf prometheus-2.54.1.linux-amd64.tar.gz

mv prometheus-2.54.1.linux-amd64/prometheus /usr/local/bin/
mv prometheus-2.54.1.linux-amd64/promtool /usr/local/bin/

mkdir -p /etc/prometheus
mkdir -p /var/lib/prometheus

mv prometheus-2.54.1.linux-amd64/consoles /etc/prometheus/
```

```
mv prometheus-2.54.1.linux-amd64/console_libraries /etc/prometheus/

useradd --no-create-home --shell /bin/false prometheus || echo "User prometheus already
exists"
chown -R prometheus:prometheus /etc/prometheus /var/lib/prometheus

echo "Creating Prometheus systemd service..."
tee /etc/systemd/system/prometheus.service > /dev/null <<EOF
[Unit]
Description=Prometheus Monitoring
After=network.target

[Service]
User=prometheus
Group=prometheus
ExecStart=/usr/local/bin/prometheus \
        --config.file /etc/prometheus/prometheus.yml \
        --storage.tsdb.path /var/lib/prometheus/ \
        --web.console.templates=/etc/prometheus/consoles \
        --web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
EOF

# --- 5. Configure 'switch' Container (SNMPv3) ---
echo "Configuring SNMPv3 in 'switch' container..."
incus exec switch -- bash -c " \
    apt-get update && \
    apt-get install -y nano snmp snmpd snmp-mibs-downloader libsnmp-dev && \
    systemctl stop snmpd && \
    net-snmp-config --create-snmpv3-user -ro -a SHA -A myAuthPass123 -x AES -X
myPrivPass456 authPrivUser \
"

echo "Creating snmpd.conf for 'switch'..."
incus file push - switch/etc/snmp/snmpd.conf <<EOF
agentAddress udp:161
sysLocation "Incus Test Lab"
sysContact Test@example.com

# SNMPv3 user access
rouser authPrivUser authPriv
EOF

echo "Starting SNMPd in 'switch'..."
incus exec switch -- systemctl start snmpd
incus exec switch -- systemctl enable snmpd
```

```
# --- 6. Configure 'AlertManager' Container ---
echo "Configuring AlertManager in 'alertmanager' container..."
incus exec alertmanager -- bash -c " \
    apt-get update && \
    apt-get install -y nano wget && \
    wget
https://github.com/prometheus/alertmanager/releases/download/v0.28.1/alertmanager-0.28.
1.linux-amd64.tar.gz && \
    tar -xvzf alertmanager-0.28.1.linux-amd64.tar.gz && \
    mkdir -p /etc/alertmanager \
"

echo "Creating alertmanager.yml..."
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# !!! IMPORTANT: Edit the email/password fields below     !!!
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
incus file push - alertmanager/etc/alertmanager/alertmanager.yml <<EOF
route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 5s
  receiver: 'email-notifications'

receivers:
  - name: 'email-notifications'
    email_configs:
      - to: 'YOUR_DESTINATION_EMAIL@gmail.com'
        from: 'YOUR_SENDING_EMAIL@gmail.com'
        smarthost: 'smtp.gmail.com:587'
        auth_username: 'YOUR_SENDING_EMAIL@gmail.com'
        auth_password: 'YOUR-16-DIGIT-APP-PASSWORD'
        auth_identity: 'YOUR_SENDING_EMAIL@gmail.com'
        require_tls: true
EOF

echo "Creating AlertManager systemd service..."
incus file push - alertmanager/etc/systemd/system/alertmanager.service <<EOF
[Unit]
Description=Prometheus Alertmanager
Wants=network-online.target
After=network-online.target

[Service]
User=root
Group=root
Type=simple
```

```
ExecStart=/root/alertmanager-0.28.1.linux-amd64/alertmanager \
  --config.file=/etc/alertmanager/alertmanager.yml \
  --cluster.listen-address=""
Restart=on-failure
RestartSec=5

[Install]
WantedBy=multi-user.target
EOF

echo "Starting AlertManager..."
incus exec alertmanager -- systemctl daemon-reload
incus exec alertmanager -- systemctl enable alertmanager
incus exec alertmanager -- systemctl start alertmanager

# --- 7. Configure 'SNMPExporter' Container ---
echo "Configuring SNMPExporter container..."
incus exec SNMPExporter -- bash -c " \
    apt-get update && \
    apt-get install -y nano wget && \
    wget
https://github.com/prometheus/snmp_exporter/releases/download/v0.29.0/snmp_exporter-0.
29.0.linux-amd64.tar.gz && \
    tar xzf snmp_exporter-0.29.0.linux-amd64.tar.gz && \
    cp snmp_exporter-0.29.0.linux-amd64/snmp_exporter /usr/local/bin/snmp_exporter && \
    chmod +x /usr/local/bin/snmp_exporter && \
    mkdir -p /etc/snmp_exporter \
"

echo "Creating snmp.yml for exporter..."
incus file push - SNMPExporter/etc/snmp_exporter/snmp.yml <<EOF
snmpv3_switch:
  walk:
    - 1.3.6.1.2.1.1       # system
    - 1.3.6.1.2.1.2       # interfaces
    - 1.3.6.1.2.1.31.1.1  # ifXTable (high capacity interfaces)
  metrics:
    - name: sysDescr
      oid: 1.3.6.1.2.1.1.1
      type: DisplayString
      help: "System description"
    - name: sysName
      oid: 1.3.6.1.2.1.1.5
      type: DisplayString
      help: "System name"
EOF

echo "Creating auth.yml for exporter..."
```

```
incus file push - SNMPExporter/etc/snmp_exporter/auth.yml <<EOF
configs:
  snmpv3_switch:
    version: 3
    username: "authPrivUser"
    security_level: "authPriv"
    auth_protocol: "SHA"
    auth_password: "myAuthPass123"
    priv_protocol: "AES"
    priv_password: "myPrivPass456"
EOF

echo "Creating SNMP Exporter systemd service..."
incus file push - SNMPExporter/etc/systemd/system/snmp_exporter.service <<EOF
[Unit]
Description=Prometheus SNMP Exporter
After=network-online.target

[Service]
User=root
Restart=on-failure
RestartSec=5
ExecStart=/usr/local/bin/snmp_exporter \
  --config.file=/etc/snmp_exporter/snmp.yml \
  --config.auth-file=/etc/snmp_exporter/auth.yml \
  --web.listen-address=0.0.0.0:9116
[Install]
WantedBy=multi-user.target
EOF

echo "Starting SNMP Exporter..."
incus exec SNMPExporter -- systemctl daemon-reload
incus exec SNMPExporter -- systemctl enable --now snmp_exporter

# --- 8. Configure Prometheus ---
echo "Configuring Prometheus to find containers..."
# We use the .incus DNS names, which is more reliable than IPs
tee /etc/prometheus/prometheus.yml > /dev/null <<EOF
global:
  scrape_interval: 15s

alerting:
  alertmanagers:
  - static_configs:
    - targets:
      - alertmanager.incus:9093

scrape_configs:
```

```
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]

  - job_name: "alertmanager"
    static_configs:
      - targets: ["alertmanager.incus:9093"]

  - job_name: "snmp-exporter"
    metrics_path: /snmp
    params:
      module: [snmpv3_switch]
      target: ["switch.incus"] # This is the SNMP agent to scrape
    static_configs:
      - targets: ["SNMPExporter.incus:9116"] # This is the exporter address
EOF

# --- 9. Restart Prometheus and Finish ---
echo "Restarting Prometheus to apply new config..."
systemctl daemon-reload
systemctl restart prometheus

echo "---"
echo "✅ Deployment Complete! ---"
echo "All services are running."
echo ""
echo "Access Prometheus at: http://<your-host-ip>:9090"
echo "Access Alertmanager at: http://<your-host-ip>:9093"
echo ""
echo "Check Prometheus targets: http://<your-host-ip>:9090/targets"
echo "  (It may take 30-60 seconds for all targets to show 'UP')"
```

Step 3: Configure Your Credentials (Critical)

```
- to: 'YOUR_DESTINATION_EMAIL@gmail.com'
      from: 'YOUR_SENDING_EMAIL@gmail.com'
      ...
      auth_username: 'YOUR_SENDING_EMAIL@gmail.com'
      auth_password: 'YOUR-16-DIGIT-APP-PASSWORD'
      auth_identity: 'YOUR_SENDING_EMAIL@gmail.com'
```

STEP 4 : Making it an executable

```
chmod +x deploy_monitoring.sh
```

STEP 5 : Running the script for automation
sudo ./deploy_monitoring.sh

STEP 6 : Running incus admin manually for the first use in another terminal

sudo incus admin init

STEP 7: Resume the installation in the primary terminal