Initializing Nodes as a Non-Root User

Overview

When adding a static or dynamic node pool, you have the option of initializing them without root privileges.

> This feature is only available for Ubuntu 18 and Ubuntu 20:
>
> - **AWS/GCP** - Ubuntu 18 and Ubuntu 20
> - **Azure** - Ubuntu 20 only
> - **Static nodes** - Ubuntu 18 and Ubuntu 20

> **enforceNonRootNodes Global Setting**
>
> Pipelines admins can use the `enforceNonRootNodes` system-level setting to enforce use of only those node pools that are configured with non-root. When this global setting is set as `true` , it takes precedence over the non-root user setting configured in the UI. Note that this is currently supported only in Ubuntu 18 and Ubuntu 20.
>
> Depending on the type of Pipelines installation, the `enforceNonRootNodes` setting can be configured as follows:
>
> - **Public chart**: If the installation was using the public chart, then change the setting in t he values.yml file.
> - **Docker**: If the installation was using Docker, then change the setting in the Runtime Configuration section in the **Pipelines System YAML**.

Prerequisites

> **Not Applicable for Dynamic Nodes**
>
> These prerequisites are not applicable for dynamic nodes as these prerequisites are automatically installed when you choose to run the build node as a non-root user.

The installation script that is generated when initializing a static node will not install any of the dependencies that would otherwise be automatically installed when you are the root user. These dependencies must be manually installed as outlined in this section.

- Following prerequisites must be installed in the build node. Since users are non-root, they won't be able to install these libraries.
  - Curl
  - jq
  - Wget
  - Tar
  - Node.js v14.17.0
  - NTP
- Swap space is pre-configured in the system.
- Custom-certificates are updated in the node manually.
- Currently, only manual node initialization is supported. So initialize the node using the same user you want to use to run other services, such as reqKick, rootless-docker, and so on.

> **reqKick**
>
> `reqKick` is the Pipelines agent that needs to run on the build node to orchestrate the build.

Setting up Rootless Docker

Perform the following steps to set up rootless docker for static nodes:

1. Login to static node as the root user and install all prerequisites mentioned above.
2. Run the following commands to install rootless docker and create a non-root used called `pipelinesRootless`

```
sudo groupadd -g 1066 pipelinesRootless
sudo adduser --system --home /home/pipelinesRootless --gid 1066 --uid 1066 --shell /bin/bash
pipelinesRootless
sudo loginctl enable-linger pipelinesRootless
sudo apt-get install -y uidmap
XDG_RUNTIME_DIR=/run/user/1066
HOME=/home/pipelinesRootless
curl -fsSL https://get.docker.com/rootless | sudo -E -u pipelinesRootless sh
```

Initializing Static Nodes as a Non-Root User

Before initilializing a static node, install the prerequisites and set up rootless docker.

To initialize a static node as a non-root user:

1. Create a static node pool. Select the **Enable running nodes with non-root users** check box when adding the node pool.
2. Add a static node and generate a manual initialization script. The script generated (for static) is slightly different for non-root.
3. SSH to the node and:
    a. Switch to `pipelinesRootless` user.
    b. Copy the generated script to `/home/pipelinesRootless`.
    c. Run the `chmod +x init.sh` command to provide executable permissions.
    `init.sh` is the name of the script.
    d. Execute the init script.

Example

```
jane@ip-10-90-104-98:/home$ su pipelinesRootless
Password:

pipelinesRootless@ip-10-90-104-98:~$ chmod +x init.sh

pipelinesRootless@ip-10-90-104-98:~$ ./init.sh
/usr/bin/curl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 10.4M  100 10.4M    0     0  66.9M      0 --:--:-- --:--:-- --:--:-- 66.9M
|___ Checking operating system...
|___ Architecture: x86_64
|___ Operating System: Ubuntu
|___ Version: 18.04
|___ wget already installed, skipping...
|___ tar already installed, skipping...
|___ jq already installed, skipping...
|___ node already installed, skipping...
|___ ntp already present, skipping...
working...
|___ Added insecure registries to docker config: { "insecure-registries": [] }
|___ Removing existing reqKick services...
|___ Booting up reqKick service...
Created symlink /home/pipelinesRootless/.config/systemd/user/multi-user.target.wants/pipelines-
reqKick.service  /home/pipelinesRootless/.config/systemd/user/pipelines-reqKick.service.
Checking if pipelines-reqKick.service is active
pipelines-reqKick.service is active
```

Limitations

Running a dynamic or static build node as a non-root user has the following limitations:

- For static nodes, the build node agent will not auto-restart on rebooting the machine. Every time the machine is rebooted, users must manually run the initialization script to re-initialize the node.
  This limitation is not applicable for dynamic nodes.
- If you run reqKick with non-root and choose the runtime as `host`, you will not be able to perform actions that a root users is allowed to do, such as installing libraries, accessing all file-systems, and so on.
- Non-root users do not have permissions to add custom CA in the build node. It becomes the responsibility of the administrators to do so.