Getting Started with QNX® in the Cloud
QNX Amazon Machine Image

Electronic edition published: February 17, 2023

Contents

Contents

About QNX Amazon Machine Image

QNX in the Cloud Amazon Machine Image (AMI) is a reference image and starting point for developers to leverage the QNX Neutrino RTOS or the QNX OS for Safety as Amazon cloud virtual machine instances. This guide describes how to set up QNX AMI instances on Amazon Elastic Compute Cloud (Amazon EC2) and quickly develop a program so you can run it on an AMI instance.

This guide assumes the following:

• You have an Amazon Web Services (AWS) account. Sign into AWS at https://aws.amazon.com/.

• You're familiar with the AWS EC2 user guides at https://docs.aws.amazon.com/ec2/index.html.

• Your organization has provided you the resources to access their AWS environment.

• On your host, you've installed the QNX Software Development Platform (SDP), which includes the QNX Momentics tool suite. QNX Momentics is a complete QNX Neutrino development environment.

• You're familiar with the *QNX Momentics IDE User's Guide*.

To:	Go to:
Connect to an existing QNX AMI instance from your host computer and the QNX Momentics IDE	Connecting to an Existing QNX AMI Instance
Create and launch a new QNX AMI instance	Launching a New QNX Amazon Machine Image Instance
Customize and save your own QNX AMI instance	Customizing the QNX Amazon Machine Image
Find the Amazon documentation about key pairs, VPCs, security groups, volumes, and so on	Managing Your Amazon Resources

Typographical conventions

Throughout this manual, we use certain typographical conventions to distinguish technical terms. In general, the conventions we use conform to those found in IEEE POSIX publications.

The following table summarizes our conventions:

Reference	Example
Code examples	`if( stream == NULL)`
Command options	`-lR`
Commands	`make`
Constants	`NULL`
Data types	`unsigned short`
Environment variables	*PATH*
File and pathnames	**/dev/null**
Function names	*exit()*
Keyboard chords	**Ctrl–Alt–Delete**
Keyboard input	`Username`
Keyboard keys	**Enter**
Program output	`login:`
Variable names	*stdin*
Parameters	*parm1*
User-interface components	**Navigator**
Window title	**Options**

We use an arrow in directions for accessing menu items, like this:

- You'll find the Other... menu item under **Perspective → Show View**.

We use notes, cautions, and warnings to highlight important messages:

Notes point out something important or useful.

CAUTION: Cautions tell you about commands or procedures that may have unwanted or undesirable side effects.

DANGER: Warnings tell you about commands or procedures that could be dangerous to your files, your hardware, or even yourself.

ALERT: Alerts tell you about commands or procedures that could expose private information or otherwise compromise security.

Note to Windows users

In our documentation, we typically use a forward slash (/) as a delimiter in pathnames, including those pointing to Windows files. We also generally follow POSIX/UNIX filesystem conventions.

Technical support

Technical assistance is available for all supported products.

To obtain technical support for any QNX product, visit the Support area on our website (https://blackberry.qnx.com/en/support). You'll find a wide range of support options.

Chapter 1
Connecting to an Existing QNX AMI Instance

This section describes how to connect to an existing QNX AMI instance from your host computer and the QNX Momentics IDE.

Checking the prerequisites for instance connections and access

Connecting to an existing instance requires that you have the proper access, which is provided by your organization.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/ and choose **Instances**.
2. Locate and open the target QNX AMI instance. On the target instance page, make sure the instance is running. Otherwise, start the instance by choosing **Instance state** > **Start instance**.
3. Check the **Security** tab to ensure the security group allows your host computer to connect.
4. Obtain the instance public DNS address and the key pair of the instance. If you didn't create the instance or the key pair, someone should provide you the key pair and your *userid* to access the instance. Record the file location where you saved the key pair.

Connecting to a QNX AMI instance with your host computer

To connect to a QNX AMI instance with your host computer, follow the instructions for using an SSH connection at https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html.

Another connection method is the EC2 serial console. Only one serial console can be active at one time for an instance and is usually reserved for the creator. Others connecting to the instance will use `ssh`.

Accessing the QNX AMI instance with the QNX Momentics IDE

To access your target instance from the QNX Momentics IDE and run programs on it, you must create a *target connection* on Momentics.

1. Open a QNX Momentics project on your host computer.
2. In the **Target Navigator** panel, create **New QNX Target**.
3. For **Hostname or IP**, enter the public IP DNS address of the QNX AMI instance.
4. For **Port**, enter the port allowed in the instance security group.

Now you can use your instance with Momentics.

You can use the QNX AMI image instance like any QNX-based system using the QNX product documentation at https://www.qnx.com/developers/docs/index.html.

For details on the Amazon environment, see the Amazon documentation at https://docs.aws.amazon.com/.

Chapter 2
Launching a New QNX Amazon Machine Image Instance

This section describes how to use the Amazon EC2 Console to create and launch a QNX AMI instance.

Launching a new QNX AMI instance

Launching a QNX AMI instance follows similar tasks in "Step 1: Launch an instance" in https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html with a QNX-specific change:

- Under **Application and OS Images (Amazon Machine Image)**, search for **QNX** and choose the most recent version as the AMI. This is the operating system (OS) image for your instance.

Some configurations to check:

- Under **Network settings**, choose **Edit**. Check that you have the appropriate VPC and security group. If you want public IP access, for **Auto-assign public IP**, select **Enable**.
- Under **Storage(volumes)**, choose **Advanced** to reveal the **EBS Volumes** details. Open the volume details and check that **Delete on termination** is set to **Yes**.

> 💡 For useful details on each item, choose **Info**.

When ready, launch your QNX AMI instance.

Booting a new QNX AMI instance

Once the new QNX AMI instance is ready, you may connect to it using the EC2 serial console or an SSH connection.

> 💡 For QNX AMI instances that are booting for the first time, you may follow the prompts to input new passwords for the `root` user and the `qnxuser` user. The first prompt will appear when you give keyboard focus to the EC2 serial console window and press ENTER.

Only one serial console can be active at one time for an instance and is usually reserved for the creator. Others connecting to the instance will use `ssh`. To allow connections from QNX Momentics IDE, run the `qconn` process.

You can use the QNX AMI image instance like any QNX-based system using the QNX product documentation at https://www.qnx.com/developers/docs/index.html.

For details on the Amazon environment, see the Amazon documentation at https://docs.aws.amazon.com/.

Chapter 3
Customizing the QNX Amazon Machine Image

The QNX AMI is a reference image generated by the QNX Image Filesystem (IFS) for developers to leverage the QNX Neutrino RTOS or the QNX OS for Safety as Amazon EC2 instances. The reference image does not contain all the libraries or binaries found on a non-AMI QNX system, and can't support the full range of functions and features found on a standard QNX target. You can't customize or tailor the QNX IFS that generates the QNX AMI reference, but the QNX AMI environment provides several customization options, including:

- managing user accounts, passwords, and groups
- adding files
- adding storage and partitioning
- extending the boot

To save your customized QNX AMI, follow the Amazon EC2 User Guide at
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-an-ami-ebs.html.

Managing user accounts, passwords, and groups

When the QNX AMI instance first boots, it prompts you to input new passwords for the `root` account and the `qnxuser` account. Once complete, you can login as either `root` or `qnxuser`.

Like any QNX-based system, you can also add users or groups you want to the system. For detailed steps, see "Managing User Accounts" in the *QNX Neutrino User's Guide* at
https://www.qnx.com/developers/docs/7.1/#com.qnx.doc.neutrino.user_guide/topic/about.html.

Adding files

Like any QNX system, you can use the Momentics IDE or the `scp` utility to copy or upload files from the host to the AMI instance, or vice versa. See the `scp` entry in the *Utilities Reference* at
https://www.qnx.com/developers/docs/7.1/#com.qnx.doc.neutrino.utilities/topic/about.html.

Currently, the QNX AMI instance has two writeable partitions. To display partition information, run `mount` in the QNX instance terminal. For example:

```
# mount
/dev/nvme0t178 on / type qnx6
ifs on / type ifs
/dev/nvme0t179 on /data type qnx6
```

The first partition **/dev/nvme0t178** is a 256MB "system" partition union mounted at **/** (slash), where you can upload (via Momentics or `scp`) your own libraries, binaries, and so on.

⚠️ **CAUTION:** Do not change the permissions of the directories under this partition. If they are changed, users will no longer be able to log in.

You can add your own update management software as appropriate. As this partition is small, it should be reserved for "system" libraries and binaries and not the application-level libraries and binaries.

The second partition **/dev/nvme0t179** is a 2.8GB writable partition mounted at **/data**. Run `ls` to list the directory content of **/** (e.g., `ls /`):

```
# ls /
bin        dev        home       proc       sbin       usr
data       etc        lib        root       tmp        var
#
```

This partition should serve as the general integration point for any or all customizations to the application-level support files such as binaries, libraries, data files, config files, and so on. Again, you can use Momentics or `scp` to upload files to this partition.

The environment variables *PATH* and *LD_LIBRARY_PATH* are set to include certain directories in the **/data** mountpoint in addition to the standard system paths, ensuring that any customizations done within the **/data** mountpoint will get picked up during the boot sequence. To view what directories are set, run `echo` with `$PATH` or `$LD_LIBRARY_PATH` in the QNX instance terminal.

For example:

```
#
# echo $PATH
/proc/boot:/sbin:/bin:/usr/bin:/usr/sbin:/usr/libexec:/data/bin:/data/sbin:/data/usr/bin:/data/usr/sbin
# echo $LD_LIBRARY_PATH
/proc/boot:/lib:/usr/lib:/lib/dll:/lib/dll/pci:/data/lib:/data/lib/dll:/data/usr/lib
#
```

Adding more storage

Initially, there is one NVME device (**nvme0**) associated with the AMI. In the QNX instance terminal, run `ls /dev` to view the devices:

```
# ls /dev
bpf           nvme0t179     ptyp3        shmem        tty          ttyp7
console       pci           ptyp4        slog         ttyp0        tymem
io-sock       pf            ptyp5        slog2        ttyp1        urandom
mem           pfil          ptyp6        socket       ttyp2        zero
mqueue        pipe          ptyp7        stderr       ttyp3
null          ptyp0         random       stdin        ttyp4
nvme0         ptyp1         sem          stdout       ttyp5
nvme0t178     ptyp2         ser1         text         ttyp6
#
```

To attach an additional volume to an instance, follow the steps in "Attach an Amazon EBS volume to an instance"" at https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-attaching-volume.html.

When the volume is attached to the QNX instance, in the QNX instance terminal, run `shutdown` to reboot the instance.

After the instance reboots, the additional associated volume appears as the next NVME device **nvme***N*, where *N* is the next available number (e.g., **nvme1**).

```
# ls /dev
bpf             nvme0t179       ptyp1           sem             stdout          ttyp5
console         nvme1           ptyp2           ser1            text            ttyp6
io-sock         nvme1t179       ptyp3           shmem           tty             ttyp7
mem             pci             ptyp4           slog            ttyp0           tymem
mqueue          pf              ptyp5           slog2           ttyp1           urandom
null            pfil            ptyp6           socket          ttyp2           zero
nvme0           pipe            ptyp7           stderr          ttyp3
nvme0t178       ptyp0           random          stdin           ttyp4
#
```

Partitioning and formatting your storage

To partition a volume (e.g., **nvme1**), follow these steps:

1. Run

   ```
   fdisk /dev/nvme1 add -t178
   ```

   which creates a type 178 partition in **nvme1**.

2. Reboot the instance with `shutdown`, then run `ls /dev` to confirm that the **nvme1t178** partition is added.

3. To format **nvme1t178**, run

   ```
   mkqnx6fs /dev/nvme1t178
   ```

4. To mount **nvme1t178** to a path, run

   ```
   mount /dev/nvme1t178 /pathname
   ```

   where *pathname* is any name you want. The mounted path is now available for use.

The QNX AMI instance can mount as many disk volumes as required.

For more on QNX IFS, see "Working with Filesystems" in the *QNX Neutrino User's Guide* at https://www.qnx.com/developers/docs/7.1/#com.qnx.doc.neutrino.user_guide/topic/fsystems.html.

Extending the boot

As mentioned previously, the IFS is not customizable. However, the QNX AMI environment provides two methods to extend the boot process:

- **startup script**

  The startup process looks for the existence of the script **/var/start_script.sh**. The IFS executes the script, integrating its code into the boot process.

- **System Launch and Monitor (SLM) configuration file**

  The startup process also looks for the existence of an SLM configuration file **/etc/slm/slm.conf**. The SLM uses this configuration file to start and manage any custom components to run during boot. See the `slm` entry in the *Utilities Reference* at https://www.qnx.com/developers/docs/7.1/#com.qnx.doc.neutrino.utilities/topic/about.html.

All extension options can be as complex or simple as you would like. If the user has their own software lifecycle management tool (to manage their application-level software), either the **start_script.sh** or SLM can be called to simply launch those custom processes such as starting the customer-specific process for the lifecycle management.

Chapter 4
Managing Your Amazon Resources

It is important to manage your resources if you are no longer using them as you may incur charges from AWS. Amazon has useful documentation about key pairs, VPCs, security groups, volumes, and so on.

Use the following table to find more about Amazon resource creation, termination, and other actions:

Topic:	Go to:
AMI lifecycle	https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ami-lifecycle.html
Instance lifecycle	https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html
Key pairs	https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html
Security group	https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html
Storage	https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volumes.html
Virtual private cloud (VPC)	https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html
VPC internet gateways	https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html

For details on the Amazon environment, see the Amazon documentation at https://docs.aws.amazon.com/ec2/index.html.