

Approche sur place 	1
Approche la plus simple    (mais pas nécessairement la plus efficace)	1
Machine VMware    (approche suggérée)	1
Machine QEMU sur Ubuntu  (approche avancée)	2
Machine QEMU sur macOS  (approche avancée partielle)	3

Approche sur place

Si vous êtes en laboratoire, alors la machine virtuelle s’y trouve déjà. Vous n’avez qu’à suivre [ces instructions](#).

Les autres approches ci-dessous permettent de travailler hors des laboratoires.

Approche la plus simple (mais pas nécessairement la plus efficace)

Vous pouvez accéder à un poste d’un laboratoire du Département via <https://bv.usherbrooke.ca/>. Il suffit de se connecter avec votre CIP (*DefaultDomain*) puis d’accéder à « Pool Dinf Labo ». Vous serez ensuite sur un poste de l’un des laboratoires physiques. Comme il y aura une couche logicielle supplémentaire, il est possible que ce ne soit parfois pas aussi rapide qu’en laboratoire.

Machine VMware (approche suggérée)

Afin de travailler localement sur votre ordinateur, vous pouvez exécuter la machine virtuelle disponible en laboratoire directement sur votre ordinateur:

- Si ce n’est pas déjà fait, installez VMware Workstation Player 16 ou 17 ( Ubuntu et  Windows), ou VMware Fusion 12 ou 13 ( macOS), que vous pouvez obtenir à partir de l’une de ces sources:

- du [site officiel de VMWare](#); ou
- des licences gratuites offertes par [l’université](#); ou
- des images fournies sur le réseau public sous

 «Logiciels > VMWare > VMware-workstation-full-17.0.0.20800274.exe»

 «Logiciels > VMWare > VMware-Workstation-Full-17.0.0-20800274.x86_64.bundle»

 «Logiciels > VMWare > VMware-Fusion-13.0.0-20802013_universal.dmg»;

- Connectez-vous au réseau de l’université. Si vous ne savez pas comment consultez [ces instructions](#).

Par exemple, vous pouvez vous connecter à `rpv.usherbrooke.ca` avec Cisco AnyConnect, puis accéder à `\\argus.dinf.fsci.usherbrooke.ca\public` dans un explorateur de fichiers sur Windows (login: `usherbrooke\votre-cip`) ou `smb://argus.dinf.fsci.usherbrooke.ca/public/` sur Ubuntu.

- Téléchargez ce fichier du réseau public vers votre ordinateur personnel:

```
/public/Cours/IFT209/Machine virtuelle ARMv8/VM-IFT209-ARM.zip
```

- Décompressez VM-IFT209-ARM.zip sur votre ordinateur;
- Lancez la machine en double-cliquant sur VM-IFT209-ARM.vmx (en vous assurant d'avoir libéré suffisamment de mémoire vive et de mémoire sur votre disque dur);
- Vous pouvez maintenant travailler, la prochaine fois vous n'aurez qu'à effectuer cette dernière étape;
- Si vous avez des problèmes, vérifiez s'il y a des mises à jour à appliquer au lecteur VMware; il se peut également que vous ayez à activer une option de virtualisation dans votre BIOS (Intel VT-X, AMD-V, Secure Virtual Machine Mode, etc.)

Machine QEMU sur Ubuntu (approche avancée)

Il est possible de se passer de la machine VMware et d'utiliser directement la machine QEMU (celle qui se lance automatiquement dans un terminal sur Ubuntu en laboratoire). À partir d'Ubuntu:

- Installez **QEMU**;
- Connectez-vous au réseau de l'université. Si vous ne savez pas comment consultez [ces instructions](#). Par exemple, vous pouvez vous connecter à `rpv.usherbrooke.ca` avec Cisco AnyConnect, puis accéder à `\\argus.dinf.fsci.usherbrooke.ca\public` dans un explorateur de fichiers sur Windows (login: `usherbrooke\votre-cip`) ou `smb://argus.dinf.fsci.usherbrooke.ca/public/` sur Ubuntu.
- Copiez ce répertoire du réseau public vers votre ordinateur personnel:

```
/public/Cours/IFT209/Machine QEMU (avancé, pour travailler sans VMware)/images/
```

- Dans un terminal de votre ordinateur, déplacez-vous dans le répertoire parent du répertoire images;
- Exécutez cette commande dans le terminal (elle se trouve sur le réseau public si vous voulez la copier plus facilement):

```
qemu-system-aarch64 -M virt -cpu cortex-a57 -m 2048 \  
-drive file=./images/arm64.img,if=none,id=blk \  
-device virtio-blk-device,drive=blk \  
-device virtio-net-device,netdev=net0 \  
-netdev user,id=net0,hostfwd=tcp::2016-:22,hostfwd=tcp::2080-:80, \  
-kernel ./images/vmlinuz-4.2.0-1-arm64 \  
-initrd ./images/initrd.img-4.2.0-1-arm64 \  
-append "root=/dev/vda2 rw console=ttyAMA0 --" \  
-nographic
```

ou celle-ci sur une plus vieille installation:

```
qemu-system-aarch64 -M virt -cpu cortex-a57 -m 2048 \  
-drive file=./images/arm64.img,if=none,id=blk \  
-device virtio-blk-device,drive=blk \  
-device virtio-net-device,netdev=net0
```

```
-net user,hostfwd=tcp::2016-:22 -redir tcp:2080::80 \
-device virtio-net-device,vlan=0 -kernel ./images/vmlinuz-4.2.0-1-arm64 \
-initrd ./images/initrd.img-4.2.0-1-arm64 \
-append "root=/dev/vda2 rw console=ttyAMA0 --" -nographic
```

- Vous devez maintenant vous connecter par SSH sur la machine QEMU (l'équivalent d'accéder aux raccourcis SOURCES sur le bureau d'Ubuntu en laboratoire, ainsi qu'au terminal login_arm sur le bureau);
- Pour ce faire, ajoutez cette ligne à la fin du fichier etc/hosts de votre ordinateur:

```
127.0.1.1      armv8
```

- Effectuez cette manipulation:

```
- Lancez «Mots de passe et clés» (seahorse)
- Allez sous Fichier > Nouveau > Clé du shell sécurisé
- Entrez une description comme «armv8» et cliquez «Créer et configurer»
- Laissez le mot de passe vide (les deux fois)
- Entrez:
      Adresse du serveur: armv8:2016
      Identifiant:      root
- Entrez le mot de passe: arm64
```

- Créez un répertoire nommé SOURCES sur votre ordinateur à l'endroit de votre choix, par ex. avec la commande «mkdir SOURCES».
 - Dans un terminal, dans le répertoire parent de SOURCES, lancez:
- ```
sshfs -p 2016 root@armv8:/root/SOURCES/ift209/ ./SOURCES/
```
- Cela peut nécessiter l'installation de sshfs, par exemple avec «sudo apt install sshfs».
- Vous avez maintenant accès au répertoire SOURCES comme s'il était nativement sur votre ordinateur;
  - La prochaine fois vous n'aurez qu'à lancer la commande qemu-system-aarch64, puis sshfs.

## Machine QEMU sur macOS 🍏 (approche avancée partielle)

Techniquement, on peut aussi utiliser QEMU directement sur macOS. Un étudiant y est arrivé en 2019 en répétant la première partie des instructions pour Ubuntu, mais cette fois en exécutant cette commande:

```
qemu-system-aarch64 -M virt -cpu cortex-a57 -m 2048 \
-drive file=./images/arm64.img,if=none,id=blk \
-device virtio-blk-device,drive=blk \
-device virtio-net-device,netdev=net0 \
-netdev user,id=net0,hostfwd=tcp::2016-:22,hostfwd=tcp::2080-:80, \
net=192.168.76.0/24,dhcpstart=192.168.76.9 \
-kernel ./images/vmlinuz-4.2.0-1-arm64 \
-initrd ./images/initrd.img-4.2.0-1-arm64 \
-append "root=/dev/vda2 rw console=ttyAMA0 --" \
-nographic
```

Il faut peut-être modifier les adresses IP, ou simplement utiliser la commande donnée pour Ubuntu. De plus, l'étudiant ne m'a pas expliqué comment il s'est connecté à la machine par SSH et je n'ai pas d'ordinateur sous macOS pour le tester. Pour y arriver, vous aurez probablement besoin de ces informations:

```
identifiant: root
mot de passe: arm64
hôte: armv8
port SSH: 2016
```