

Boot2Container

la flexibilité des conteneurs sur bare metal

Vincent Autefage

vincent.autefage@u-bordeaux.fr

Martin Roukala

martin.roukala@mupuf.org



Contexte – Automatisation de tests bas niveau




Besoin de **Valve** d'automatiser l'exécution de suites de tests bas niveaux (e.g. module, noyau, pilote) sur une grande variété de machines physiques dont la **Steam Deck** dans le but de faire tourner des jeux **Steam** sur Linux.

Nombreux projets concernés hébergés sur des forges logicielles distinctes.




Une unique ferme de machines physiques dédiées à l'ensemble de ces tests.

The Valve logo, consisting of the word "VALVE" in white, uppercase, sans-serif font, centered within a red rectangular background.

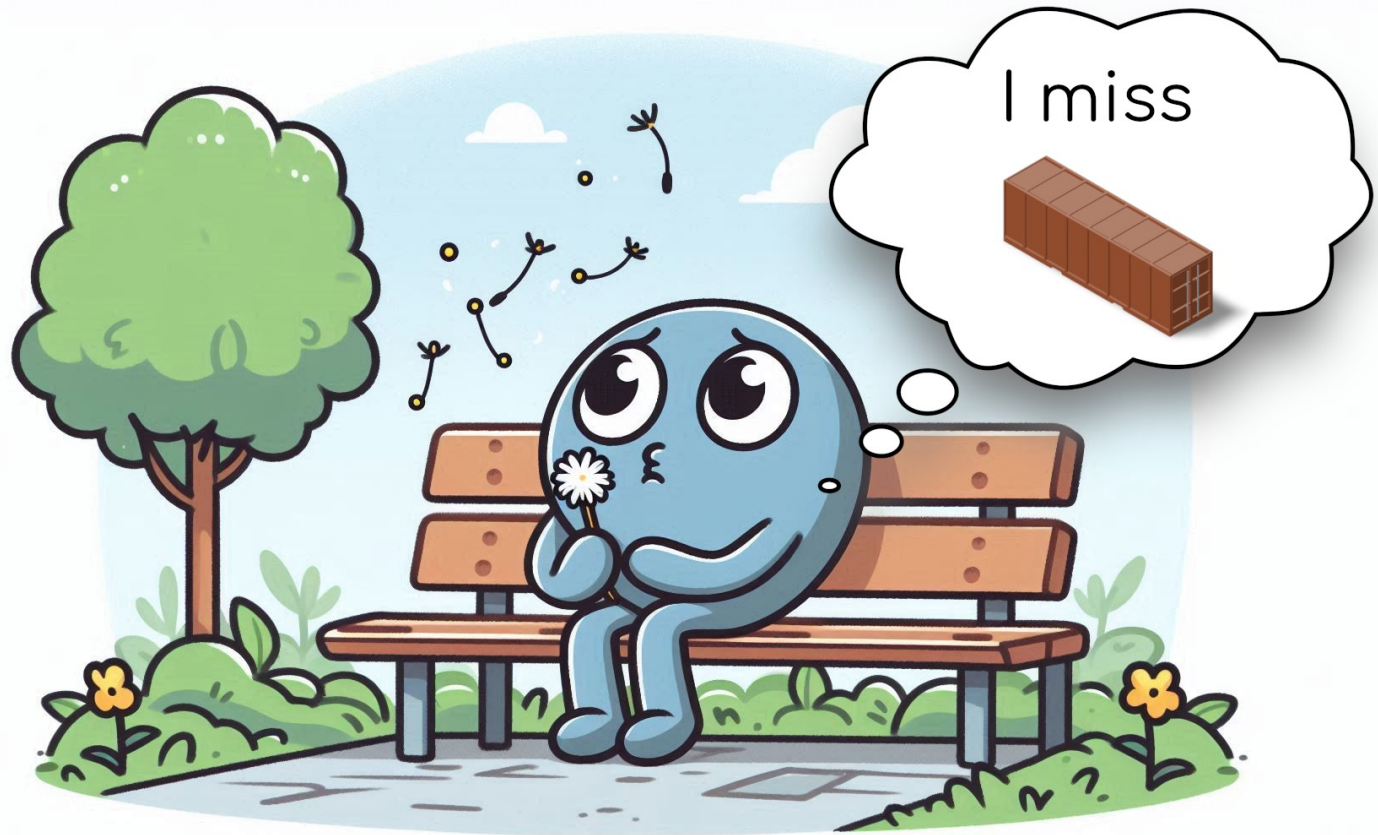
Contexte – Stratégie d'intégration continue

-  Intégration continue conventionnelle
-  Système d'auto-configuration
-  Réinstallation / flash du système

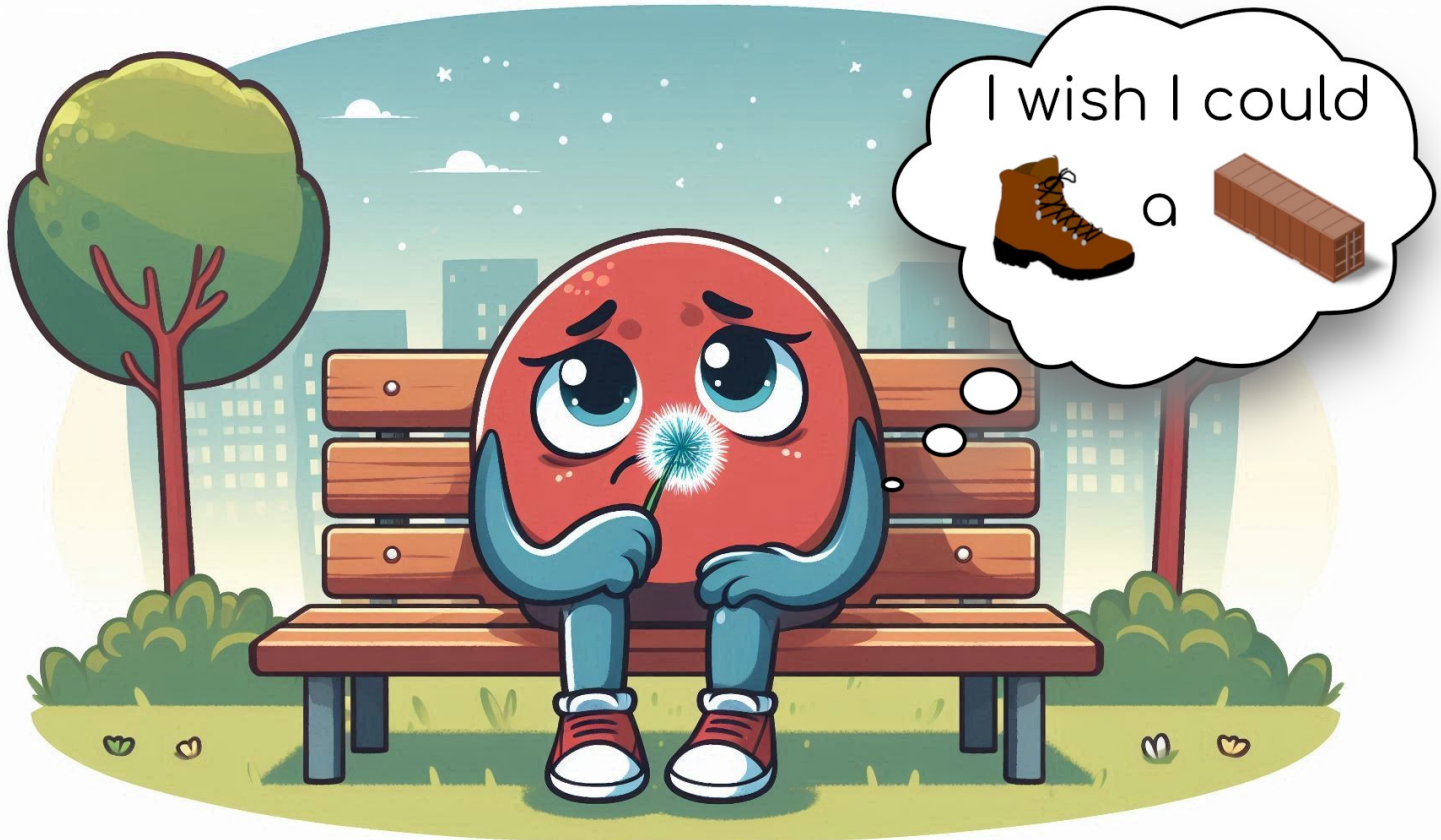


-  Temps d'installation / configuration
-  Complexité de mise en place
-  Manque d'immuabilité

Contexte – Origin Story



Contexte – Origin Story



b2c – boot2container

Projet libre sous Licence MIT amorcé en 2021

Composé d'un **initrd** de ~ 20 Mo et d'un **noyau Linux** générique

Codé en **shell script** ainsi qu'en **Go**

Permet de récupérer et d'instancier des conteneurs avec **Podman**

Configurable au travers de la ligne de commande du noyau

Disponible pour les architectures **x86_64**, **ARMv6**, **ARM64** et **RISCV64**

Applicable aux machines physiques ou virtuelles sans OS ni stockage



boot



linux



b2c



podman

Utilisation sur VM

- > `wget https://gitlab.freedesktop.org/.../linux-x86_64-qemu`
- > `wget https://gitlab.freedesktop.org/.../initramfs.linux_amd64.cpio.xz`
- > `qemu-system-x86_64 -m 1G -accel kvm \`
`-kernel linux-x86_64-qemu \`
`-initrd initramfs.linux_amd64.cpio.xz \`
`-append 'b2c.run="-it docker.io/alpine:3.20"'`

L'exemple ci-dessus revient peu ou prou à :
`podman run --privileged -it docker.io/alpine:3.20`

Démarrage sur QEMU

Amorcer une Alpine
Linux sur une machine
virtuelle QEMU sans OS
ni support de stockage



qemu



b2c



alpine

Synopsis

Général

Toute les commandes de b2c possède la syntaxe suivante :

```
b2c.<command>[=<parameters...>]
```

Exemples :

```
b2c.run = "-it registry.fake/alpine"
```

```
b2c.hostname = plop
```

```
b2c.keymap = fr
```

```
b2c.login = registry.fake,\  
            username=kovalsik,\  
            password=cacahuètes
```


Synopsis

Instances

Il existe 3 types d'instances :

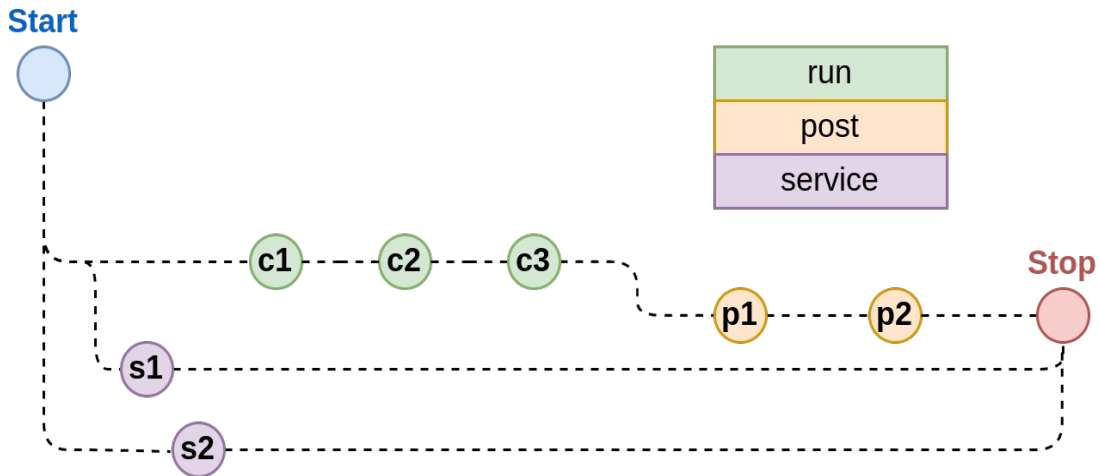
`b2c.run`, `b2c.run_post` et `b2c.run_service`

Exemple :

`b2c.run_service=s1` `b2c.run_service=s2`

`b2c.run=c1` `b2c.run=c2` `b2c.run=c3`

`b2c.run_post=p1` `b2c.run_post=p2`



Synopsis

Couche Réseau

Par défaut, les conteneurs sont lancés avec le flag `--network=host`

b2c lance un client DHCP sur toutes les interfaces Ethernet disponibles.

Il est néanmoins possible de modifier ce comportement.

Exemple :

```
b2c.ntp_peer = 0.fr.pool.ntp.org
```

```
b2c.iface = eth0,dhcp,nameserver=9.9.9.9
```

```
b2c.iface = eth1,address=192.168.42.1/24
```

```
b2c.iface = eth2,dhcp,forward
```

Synopsis

Cache

b2c s'exécute intégralement en mémoire vive, images de conteneurs comprises.

Un système de cache configurable peut être mis en place pour stocker les images de conteneurs et les volumes sur :

- ✓ stockage de masse (e.g. nvme, sata)
- ✓ stockage réseau (e.g. nfs, nbd)

Exemples :

```
b2c.cache_device = auto
```

```
b2c.swap = 8G
```

```
b2c.filesystem = my_nfs,type=nfs,src=10.0.0.1:/
```

```
b2c.cache_device = my_nfs
```

Synopsis

Volumes

b2c permet de définir des volumes locaux ou distants avec ou sans chiffrement.

Les volumes locaux sont par défaut stockés dans le périphérique de cache.

Exemple :

```
b2c.minio = srv-s3, \  
  https://s3.fake,${access_key},${secret_key}
```

```
b2c.volume = vol-s3, \  
  mirror=srv-s3/my_bucket, \  
  pull_on=pipeline_start, \  
  push_on=container_end
```

```
b2c.volume = vol-data
```

```
b2c.run = "-it -v vol-s3:/mnt/s3 \  
  -v vol-data:/mnt/data \  
  docker.io/alpine:3.20"
```

Utilisation sur bare metal

```
#!ipxe
```

```
kernel https://gitlab.freedesktop.org/.../linux-x86_64 \
initrd=b2c \
b2c.hostname=web \
b2c.iface=eth0,dhcp \
b2c.iface=eth1,dhcp \
b2c.cache_device=auto \
b2c.filesystem=fs-web,type=nfs,src=w.fake:/web,opt=ro \
b2c.volume=vol-web,filesystem=fs-web \
b2c.volume=vol-log \
b2c.run="-v vol-web:/usr/share/nginx/html:ro \
-v vol-log:/var/log \
docker.io/nginx:latest"
```

```
initrd --name b2c https://gitlab.freedesktop.org/.../initramfs.linux_amd64.cpio.xz
boot
```

Démarrage avec iPXE

Amorcer un serveur Nginx sur une **machine bare metal** avec cache mais sans OS via le chargeur d'amorçage iPXE



ipxe



b2c



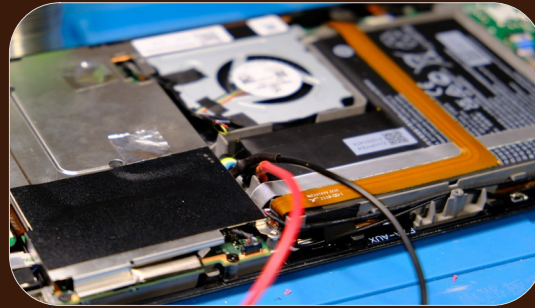
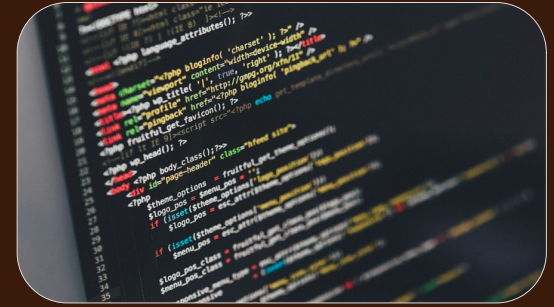
nginx

Exemples d'utilisation concrète

Affichage embarqué



Mode examen



Intégration continue bas niveau



www.u-bordeaux.fr



@univbordeaux



@universitedebordeaux



@univbordeaux



@universitedebordeaux



Appli mobile U&me



@univbordeaux

Merci de votre attention

université
de **BORDEAUX**