

Checkpoint/Restore in Openshift

Franjo Piškur

Dario Sindičić

APIS IT

- More than 60 years of experience in IT
- > 550 employees
- The greatest national IT integrator
- IT solutions for state government
- IT solutions for local government
- Contribution to the development of digital society
- collocation of IT equipment and data in the two most modern data centers



Proof of work

- Efficient Container Image Distribution for Multi-Tenant Kubernetes Clusters – N. Momčilović, D. Sindičić
- Kubernetes Discovery Chaotic Cache and How to Speed Up Your Velero Backups - D.Sindičić => PR #8166
- Issues introducing on-prem Cloud IDE in an organization – D.Posarić, D.Sindičić



Checkpoint/Restore



Issue #3949

- Technology that saves the live state of a running process or container into collection of files on disk, allowing it to be later restored to that exact same state
- Fast recovery from failures, maintenance (like reboots and kernel upgrades)
 - Containers: Checkpoint/Restore In Userspace (CRIU)
 - Virtual Machines: Platforms like Hyper-V
 - Applications: Coordinated Restore at Checkpoint (CRaC)



CRIU

- Pavel Emelyanov – creator of OpenVZ
- Initial release - 15 July 2011
- What does it do?
- Save/restore process memory, registers, open files, sockets, and more.
- Works in userspace



2015
1.13



2018
0.10



2022
1.22



Adrian Reber
RedHat



Enable Checkpoint/Restore in Openshift

4.12 <= Openshift <=4.16

```
spec:
  customNoUpgrade:
    enabled:
      - ContainerCheckpoint
  featureSet: CustomNoUpgrade
```



Knowledgebase -7008477

4.17 <= Openshift GA, generally available

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: kube-apiserver-checkpoints
rules:
- apiGroups:
  - ""
  resources:
  - nodes/checkpoint
verbs:
- get
- create
```



Let's create a snapshot



- Example

```
curl -k -X POST --header "Authorization: Bearer $TOKEN"  
https://\$OPENSHIFT\_API/api/v1/nodes/\$WORKER\_NAME/proxy/checkpoint/  
\$NAMESPACE/\$POD\_NAME/\$CONTAINER\_NAME
```

```
{  
  "items": [  
    "/var/lib/kubelet/checkpoints/checkpoint-  
$POD_NAME_$NAMESPACE-$CONTAINER_NAME-$TIME.tar"  
  ]  
}
```



Container snapshot

- `./checkpointctl inspect /home/openshift/jenkins.tar --files`
- `podman run --name jenkins -p 8080:8080 -d jenkins`
- `podman container checkpoint jenkins`




```

Process tree
└─ [1] go-init
   └─ Open files
      ├── [REG 0] /dev/null
      ├── [PIPE 1] pipe[308233808]
      ├── [PIPE 2] pipe[308233809]
      ├── [EVENTPOLL 4] EVENTPOLL.14
      ├── [PIPE 5] pipe[308231795]
      ├── [PIPE 6] pipe[308231795]
      ├── [cwd] /
      └── [root] /
   └─ [1364] sh
      └─ Open files
         ├── [REG 0] /dev/null
         ├── [cwd] /var/lib/jenkins/jobs/[REDACTED]/jobs/cns/workspace
         └── [root] /
      └─ [1365] sh
         └─ Open files
            ├── [REG 0] /dev/null
            ├── [cwd] /var/lib/jenkins/jobs/[REDACTED]/jobs/cns/workspace
            └── [root] /
         └─ [1371] sleep
            └─ Open files
               ├── [REG 0] /dev/null
               ├── [cwd] /var/lib/jenkins/jobs/[REDACTED]/jobs/cns/workspace
               └── [root] /
         └─ [1366] sh
            └─ Open files
               ├── [REG 0] /dev/null
               ├── [REG 1] /var/lib/jenkins/jobs/[REDACTED]/jobs/cns/workspace@tmp/durable-31da3761/jenkins-log.txt
               ├── [REG 2] /var/lib/jenkins/jobs/[REDACTED]/jobs/cns/workspace@tmp/durable-31da3761/jenkins-log.txt
               ├── [REG 255] /var/lib/jenkins/jobs/[REDACTED]/jobs/cns/workspace@tmp/durable-31da3761/script.sh
               ├── [cwd] /var/lib/jenkins/jobs/[REDACTED]/jobs/cns/workspace
               └── [root] /
            └─ [1369] sleep
               └─ Open files
                  ├── [REG 0] /dev/null
                  ├── [REG 1] /var/lib/jenkins/jobs/[REDACTED]/jobs/cns/workspace@tmp/durable-31da3761/jenkins-log.txt
                  ├── [REG 2] /var/lib/jenkins/jobs/[REDACTED]/jobs/cns/workspace@tmp/durable-31da3761/jenkins-log.txt
                  ├── [cwd] /var/lib/jenkins/jobs/[REDACTED]/jobs/cns/workspace
                  └── [root] /

```



Displaying container checkpoint tree view from /var/lib/kubelet/checkpoints/checkpoint-test-7dbcd6bbd9-
iner-2025-09-07T14:44:54Z.tar

container

- Image: image-registry.openshift-image-registry.svc:5000/[REDACTED]/[REDACTED]:latest
- ID: 9cf7a131962cf796b5e82a188bd25b06b1389ef6766abcbba5bce4fb1cbcee27
- Runtime: runc
- Created: 2025-09-07T14:23:53.378203116Z
- Checkpointed: 2025-09-07T14:44:54Z
- Engine: CRI-O
- IP: 10.254.6.251
- Checkpoint size: 9.8 MiB
 - Memory pages size: 9.7 MiB
- Root FS diff size: 786.9 MiB
- Process tree
 - [1] [REDACTED]-agent
 - Open sockets
 - [TCP (ESTABLISHED)] 10.254.6.251:45936 -> 10.1.55.110:8080 (↑ 45.0 KB ↓ 128.0 KB)



Looking into dumps

JWT Token

```
# strings pages*.img | egrep -o 'eyJ[A-Za-z0-9_-]{10,}\.[A-Za-z0-9_-]{10,}\.[A-Za-z0-9_-]{10,}' | sort -u
eyJhbGciOiJSUzI1NiIsI.....9lIqPi1

# crit show files.img | jq '.entries[]
  | select(.reg?.name? // "" | contains("secret.key")) | .reg.name'
"/var/lib/jenkins/secret.key"

# hexdump -C pages-*.img | more
016954b0  0a a1 07 00 40 00 00 00  63 62 64 39 65 36 37 34  |....@...cbd9e674|
016954c0  30 61 33 37 61 65 63 39  31 66 34 31 34 39 62 38  |0a37aec91f4149b8|
016954d0  36 35 62 62 63 39 32 61  63 37 30 33 36 61 62 30  |65bbc92ac7036ab0|
016954e0  32 31 35 32 63 65 65 34  35 61 32 65 33 65 31 39  |2152cee45a2e3e19|
016954f0  32 65 31 34 32 64 39 38  01 68 1e 4a 44 00 00 00  |2e142d98.h.JD...|
01695500  81 6c 06 00 a2 2a 61 ea  d6 4c a7 cd 00 00 00 00  |.l...*a..L.....|
```

Jenkins
secret key



How does it work?

P jenkins-58f9d5dd5d-f7ft5

Jenkins container

```
go-init(1)-+-java(15)-+-{java}(2613)
|               |-{java}(2614)
|               |-{java}(2615)
|               |-{java}(2616)
|               |-{java}(21444)
|               `-{java}(32240)
|-{go-init}(7)
|-{go-init}(14)
`-{go-init}(586)
```

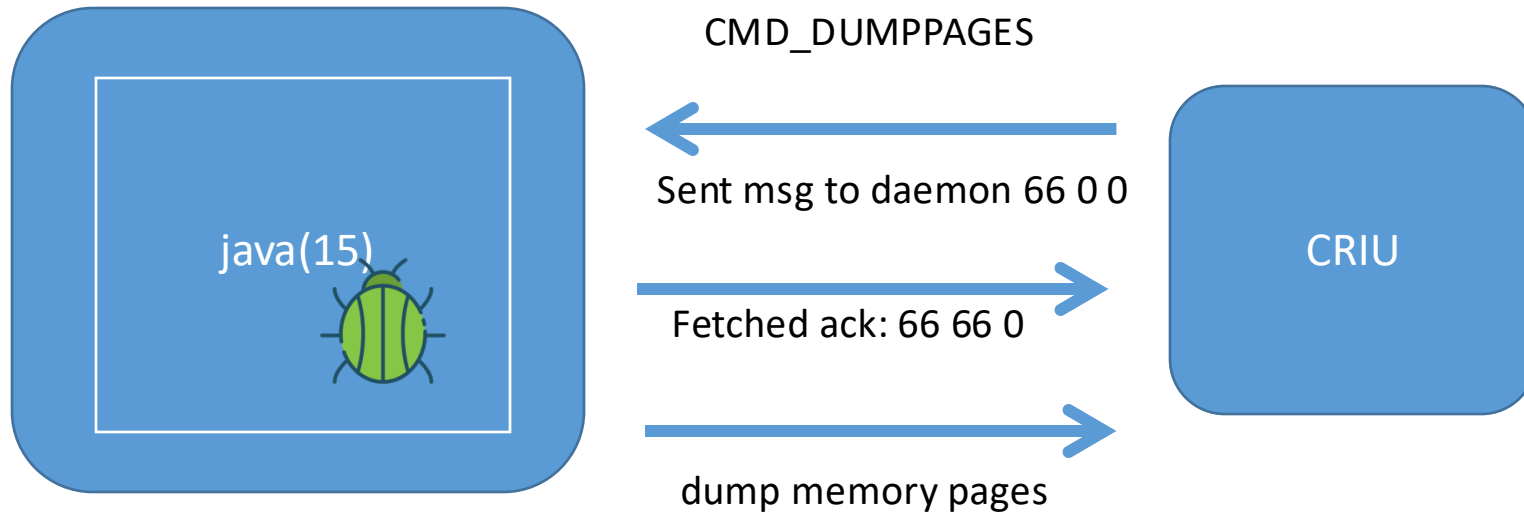
- Capture the process tree
- Send freeze to all processes
 - freezer.state → FROZEN
- Save registry state of all processes - PTRACE_GETREGSET
- Inject memory pointer inside of the memory of a process - PTRACE_POKEDATA
- Alter the registry of the next instruction to point to the injected pointer - PTRACE_SETREGSET (rip)
- Parasite code (collects all data from the inside) - PTRACE_CONT



How does it work?

criu/include/parasite.h  

 jenkins-58f9d5dd5d-f7ft5



```
enum {  
    PARASITE_CMD_DUMP_THREAD = PARASITE_USER_CMDS,  
    PARASITE_CMD_MPROTECT_VMAS,  
    PARASITE_CMD_DUMPPAGES,  
  
    PARASITE_CMD_DUMP_SIGACTS,  
    PARASITE_CMD_DUMP_ITIMERS,  
    PARASITE_CMD_DUMP_POSIX_TIMERS,  
    PARASITE_CMD_DUMP_MISC,  
    PARASITE_CMD_DRAIN_FDS,  
    PARASITE_CMD_GET_PROC_FD,  
    PARASITE_CMD_DUMP_TTY,  
    PARASITE_CMD_CHECK_VDSO_MARK,  
    PARASITE_CMD_CHECK_AIOS,  
    PARASITE_CMD_DUMP_CGROUP,  
  
    PARASITE_CMD_MAX,  
};
```



Restore

```
# newcontainer=$(buildah from scratch)
# buildah add $newcontainer checkpoint.tar /
# buildah config \
    --annotation=io.kubernetes.cri-o.annotations.checkpoint.name=c_name \
    $newcontainer
# buildah commit $newcontainer checkpoint-image:latest
# buildah rm $newcontainer
# buildah push ...
```

Snapshot



OCI Image





```
Sun Sep 7 12:19:42 UTC 2025
[Pipeline] sh
+ sleep 1
[Pipeline] echo
Iteration: 36
[Pipeline] sh
+ date
Sun Sep 7 12:19:44 UTC 2025
[Pipeline] sh
+ sleep 1
[Pipeline] echo
Iteration: 37
[Pipeline] sh
+ date
Sun Sep 7 12:27:03 UTC 2025
[Pipeline] sh
+ sleep 1
[Pipeline] echo
Iteration: 38
[Pipeline] sh
+ date
Sun Sep 7 12:27:05 UTC 2025
[Pipeline] sh
+ sleep 1
[Pipeline] echo
Iteration: 39
[Pipeline] sh
+ date
Sun Sep 7 12:27:06 UTC 2025
[Pipeline] sh
+ sleep 1
```

```
pipeline {
  agent any
  stages {
    stage('Hello') {
      steps {
        echo 'Hello World';
        script {
          for (int i=1;i<=100;i++) {
            echo "Iteration: ${i}";
            sh 'date'
            sh 'sleep 1'
          }
        }
      }
    }
  }
}
```





Pitanja