ATTACK DEFENSE

by PentesterAcademy

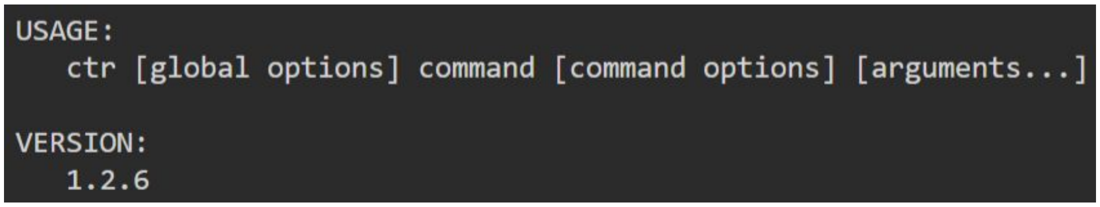
Name	Containerd Basics Lab
URL	https://attackdefense.com/challengedetails?cid=1450
Type	Docker Security : Container Basics

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

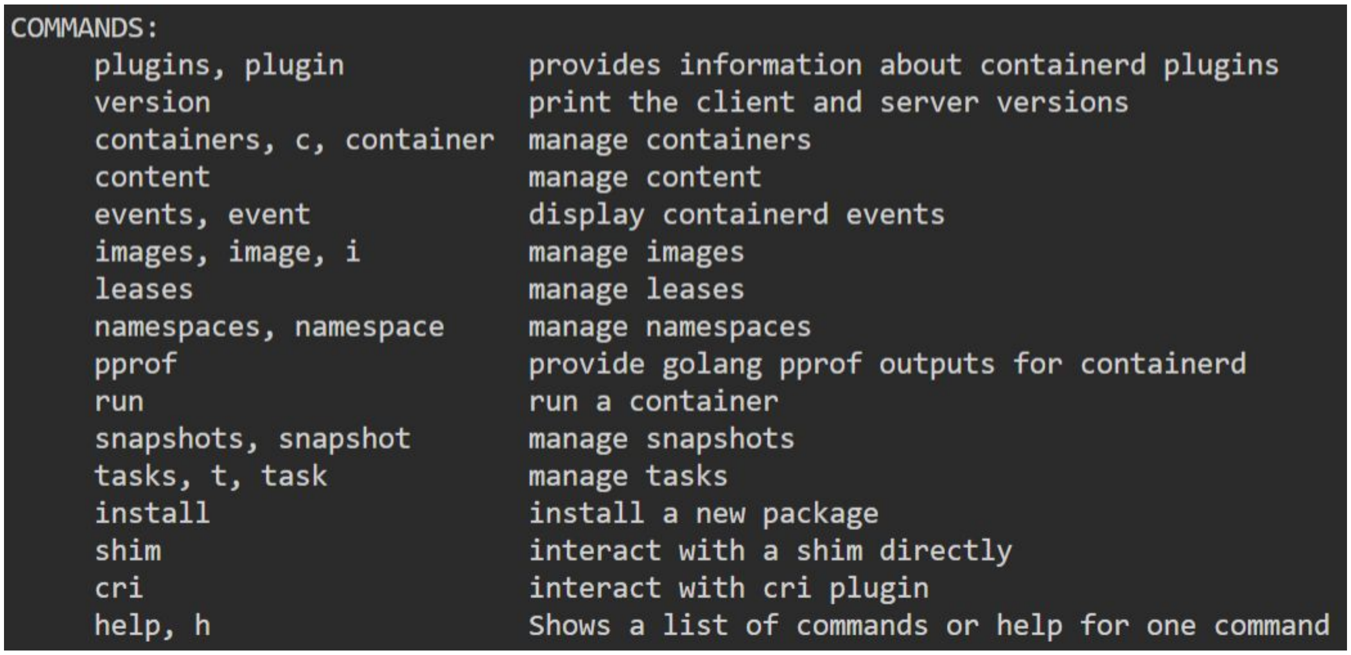
Objective: Learn basic CTR (Containerd Client) commands.

Check command help

Command: ctr



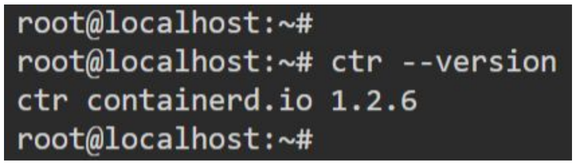
```
USAGE:
   ctr [global options] command [command options] [arguments...]

VERSION:
   1.2.6
```



```
COMMANDS:
     plugins, plugin            provides information about containerd plugins
     version                    print the client and server versions
     containers, c, container   manage containers
     content                    manage content
     events, event              display containerd events
     images, image, i           manage images
     leases                     manage leases
     namespaces, namespace      manage namespaces
     pprof                      provide golang pprof outputs for containerd
     run                        run a container
     snapshots, snapshot        manage snapshots
     tasks, t, task             manage tasks
     install                    install a new package
     shim                       interact with a shim directly
     cri                        interact with cri plugin
     help, h                    Shows a list of commands or help for one command
```

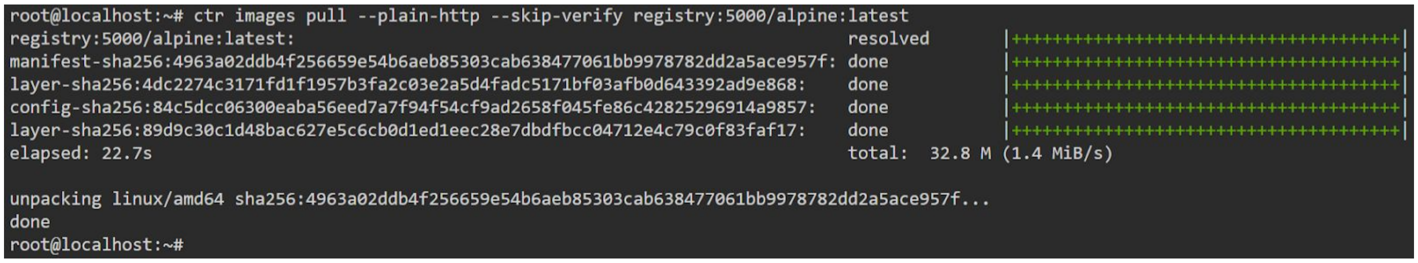
1. Check version

Command: ctr --version


```
root@localhost:~#
root@localhost:~# ctr --version
ctr containerd.io 1.2.6
root@localhost:~#
```

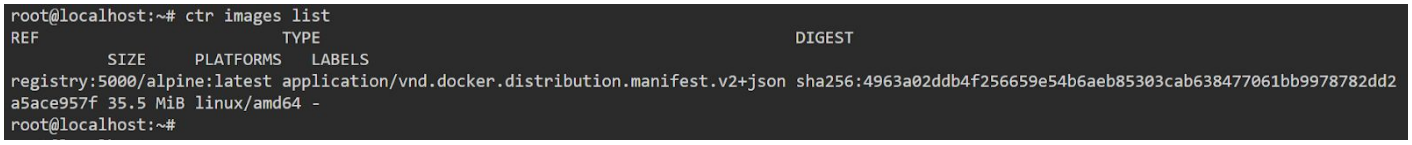
2. Pulling image

Command: ctr images pull --skip-verify --plain-http registry:5000/alpine:latest


```
root@localhost:~# ctr images pull --plain-http --skip-verify registry:5000/alpine:latest
registry:5000/alpine:latest:                                                      resolved       |++++++++++++++++++++++++++++++++++++++|
manifest-sha256:4963a02ddb4f256659e54b6aeb85303cab638477061bb9978782dd2a5ace957f: done           |++++++++++++++++++++++++++++++++++++++|
layer-sha256:4dc2274c3171fd1f1957b3fa2c03e2a5d4fadc5171bf03afb0d643392ad9e868:    done           |++++++++++++++++++++++++++++++++++++++|
config-sha256:84c5dcc06300eaba56eed7a7f94f54cf9ad2658f045fe86c42825296914a9857:   done           |++++++++++++++++++++++++++++++++++++++|
layer-sha256:89d9c30c1d48bac627e5c6cb0d1ed1eec28e7dbdfbcc04712e4c79c0f83faf17:    done           |++++++++++++++++++++++++++++++++++++++|
elapsed: 22.7s                                                                    total:  32.8 M (1.4 MiB/s)

unpacking linux/amd64 sha256:4963a02ddb4f256659e54b6aeb85303cab638477061bb9978782dd2a5ace957f...
done
root@localhost:~#
```

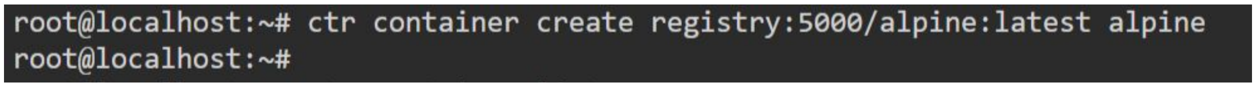
3. List images

Command: ctr images list


```
root@localhost:~# ctr images list
REF                         TYPE                                                   DIGEST
         SIZE     PLATFORMS   LABELS
registry:5000/alpine:latest application/vnd.docker.distribution.manifest.v2+json sha256:4963a02ddb4f256659e54b6aeb85303cab638477061bb9978782dd2
a5ace957f 35.5 MiB linux/amd64 -
root@localhost:~#
```

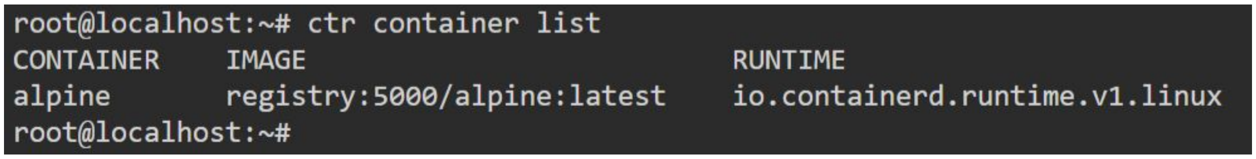
4. Create Container

Command: ctr container create registry:5000/alpine:latest alpine


```
root@localhost:~# ctr container create registry:5000/alpine:latest alpine
root@localhost:~#
```

5. List Containers

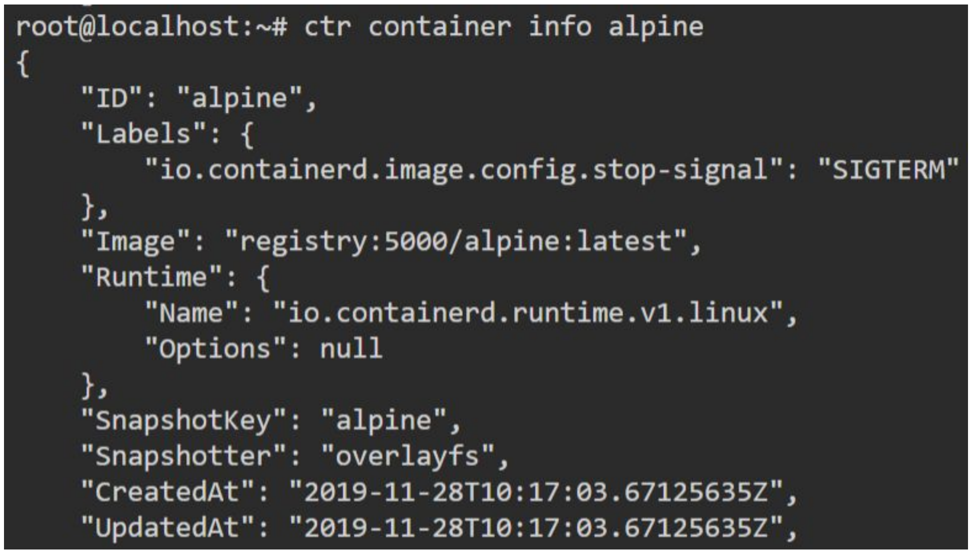
Command: ctr container list



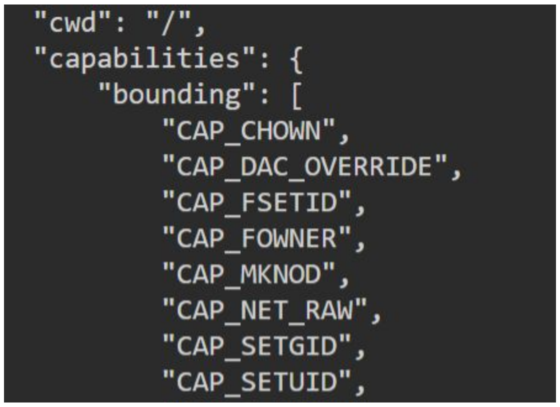
```
root@localhost:~# ctr container list
CONTAINER    IMAGE                            RUNTIME
alpine       registry:5000/alpine:latest      io.containerd.runtime.v1.linux
root@localhost:~#
```

6. Check container info

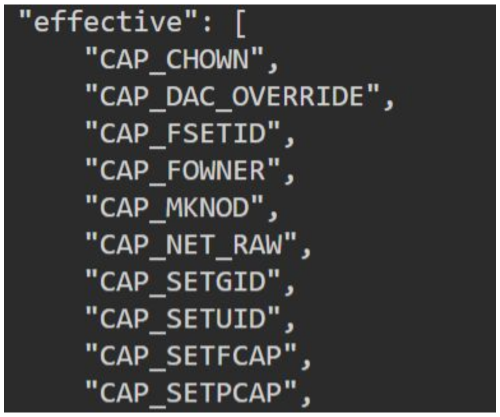
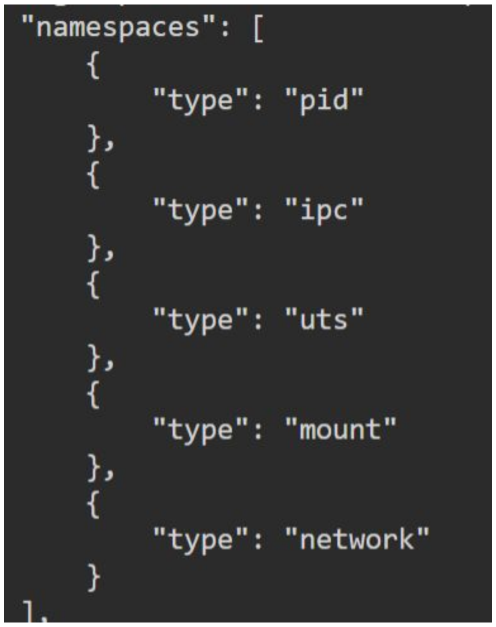
Command: ctr container info



```
root@localhost:~# ctr container info alpine
{
    "ID": "alpine",
    "Labels": {
        "io.containerd.image.config.stop-signal": "SIGTERM"
    },
    "Image": "registry:5000/alpine:latest",
    "Runtime": {
        "Name": "io.containerd.runtime.v1.linux",
        "Options": null
    },
    "SnapshotKey": "alpine",
    "Snapshotter": "overlayfs",
    "CreatedAt": "2019-11-28T10:17:03.67125635Z",
    "UpdatedAt": "2019-11-28T10:17:03.67125635Z",
```

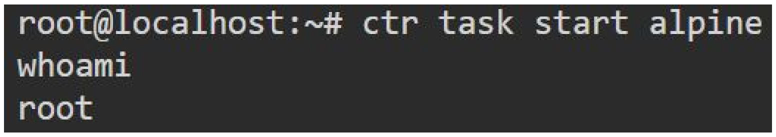


```
"cwd": "/",
"capabilities": {
    "bounding": [
        "CAP_CHOWN",
        "CAP_DAC_OVERRIDE",
        "CAP_FSETID",
        "CAP_FOWNER",
        "CAP_MKNOD",
        "CAP_NET_RAW",
        "CAP_SETGID",
        "CAP_SETUID",
```


```
"effective": [
    "CAP_CHOWN",
    "CAP_DAC_OVERRIDE",
    "CAP_FSETID",
    "CAP_FOWNER",
    "CAP_MKNOD",
    "CAP_NET_RAW",
    "CAP_SETGID",
    "CAP_SETUID",
    "CAP_SETFCAP",
    "CAP_SETPCAP",
```

```
"namespaces": [
    {
        "type": "pid"
    },
    {
        "type": "ipc"
    },
    {
        "type": "uts"
    },
    {
        "type": "mount"
    },
    {
        "type": "network"
    }
],
```

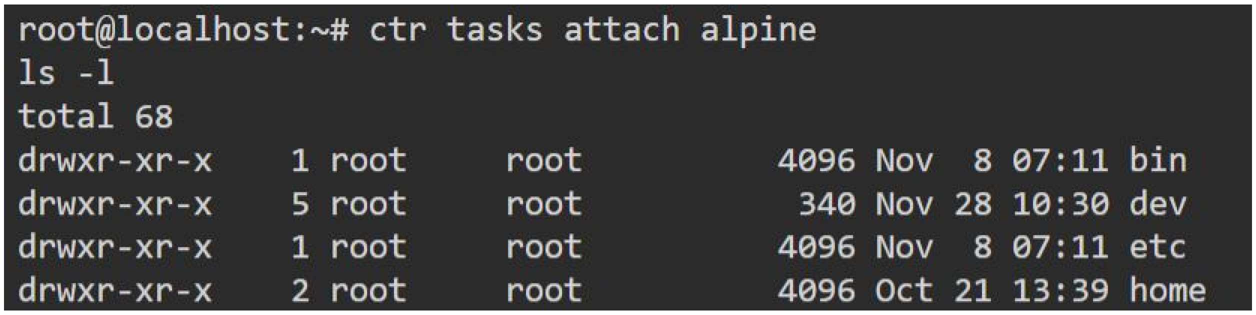
7. Start a task

Command: ctr task start alpine


```
root@localhost:~# ctr task start alpine
whoami
root
```

8. Attach to a task

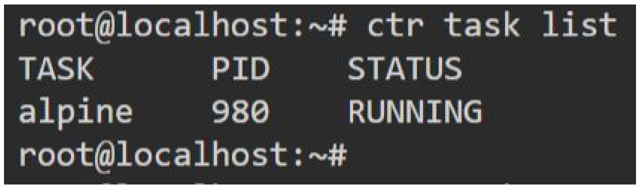
Command: ctr task attach alpine



```
root@localhost:~# ctr tasks attach alpine
ls -l
total 68
drwxr-xr-x    1 root     root          4096 Nov  8 07:11 bin
drwxr-xr-x    5 root     root           340 Nov 28 10:30 dev
drwxr-xr-x    1 root     root          4096 Nov  8 07:11 etc
drwxr-xr-x    2 root     root          4096 Oct 21 13:39 home
```

9. List tasks

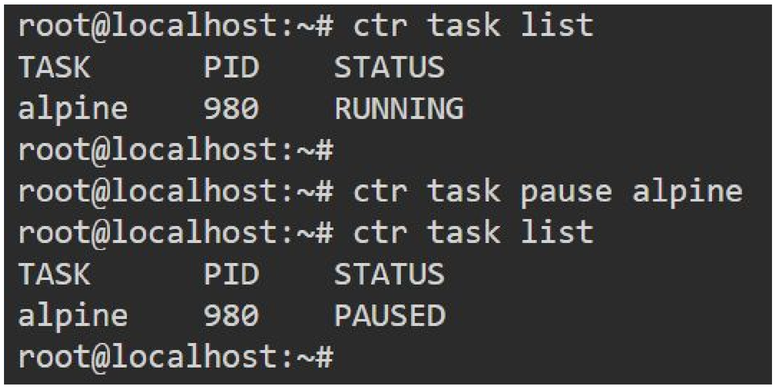
Command: ctr task list



```
root@localhost:~# ctr task list
TASK      PID    STATUS
alpine    980    RUNNING
root@localhost:~#
```

10. Pause a task

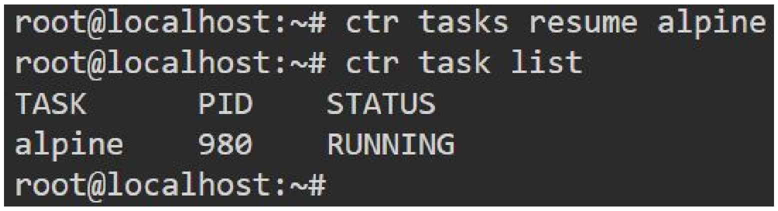
Command: ctr task pause alpine



```
root@localhost:~# ctr task list
TASK      PID    STATUS
alpine    980    RUNNING
root@localhost:~#
root@localhost:~# ctr task pause alpine
root@localhost:~# ctr task list
TASK      PID    STATUS
alpine    980    PAUSED
root@localhost:~#
```

11. Resume a task

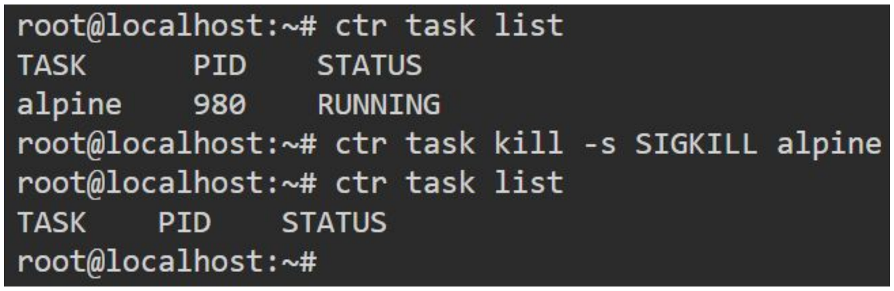
Command: ctr tasks resume alpine



```
root@localhost:~# ctr tasks resume alpine
root@localhost:~# ctr task list
TASK      PID    STATUS
alpine    980    RUNNING
root@localhost:~#
```

12. Kill a task

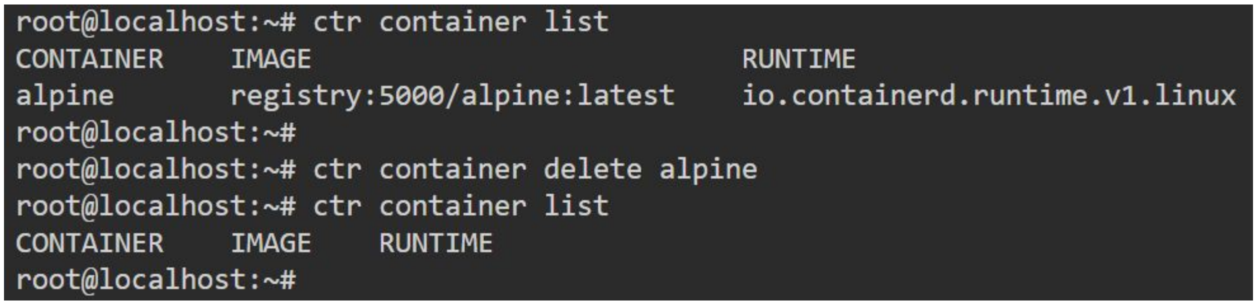
Command: ctr task kill -s SIGKILL alpine



```
root@localhost:~# ctr task list
TASK      PID    STATUS
alpine    980    RUNNING
root@localhost:~# ctr task kill -s SIGKILL alpine
root@localhost:~# ctr task list
TASK    PID    STATUS
root@localhost:~#
```

13. Delete a container

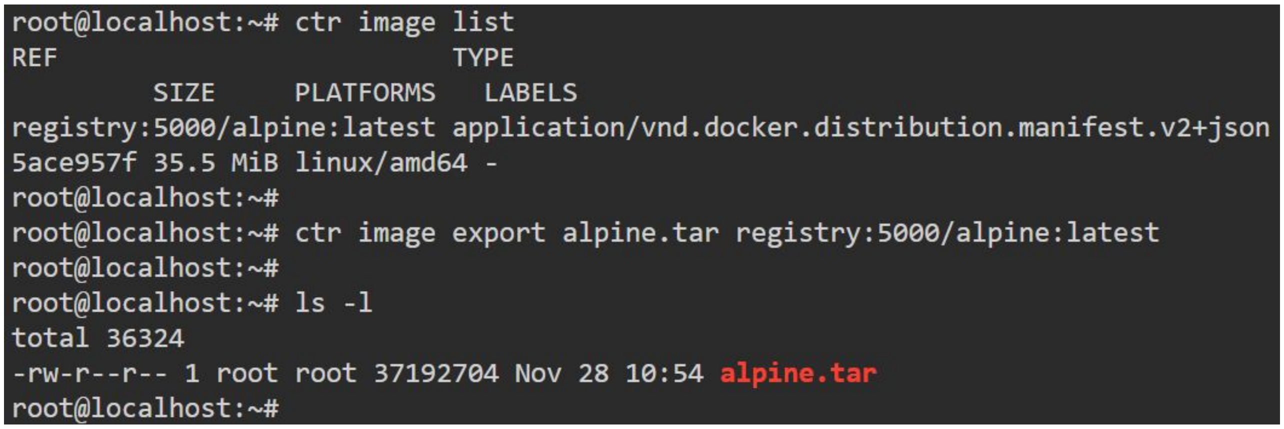
Command: ctr container delete alpine



```
root@localhost:~# ctr container list
CONTAINER    IMAGE                          RUNTIME
alpine       registry:5000/alpine:latest    io.containerd.runtime.v1.linux
root@localhost:~#
root@localhost:~# ctr container delete alpine
root@localhost:~# ctr container list
CONTAINER    IMAGE    RUNTIME
root@localhost:~#
```

14. Export image as tar archive

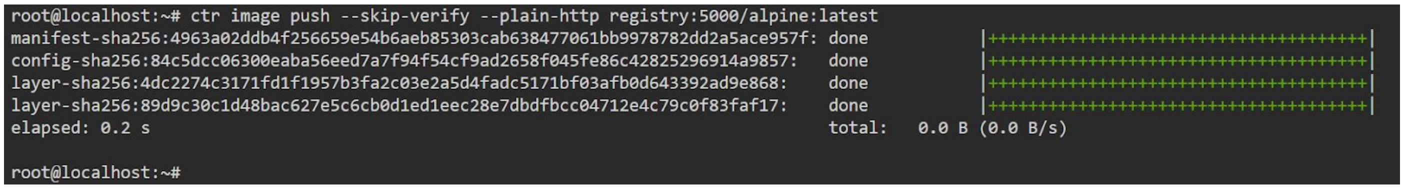
Command: ctr image export alpine.tar registry:5000/alpine:latest



```
root@localhost:~# ctr image list
REF                            TYPE
         SIZE     PLATFORMS    LABELS
registry:5000/alpine:latest application/vnd.docker.distribution.manifest.v2+json
5ace957f 35.5 MiB linux/amd64 -
root@localhost:~#
root@localhost:~# ctr image export alpine.tar registry:5000/alpine:latest
root@localhost:~#
root@localhost:~# ls -l
total 36324
-rw-r--r-- 1 root root 37192704 Nov 28 10:54 alpine.tar
root@localhost:~#
```

15. Push image

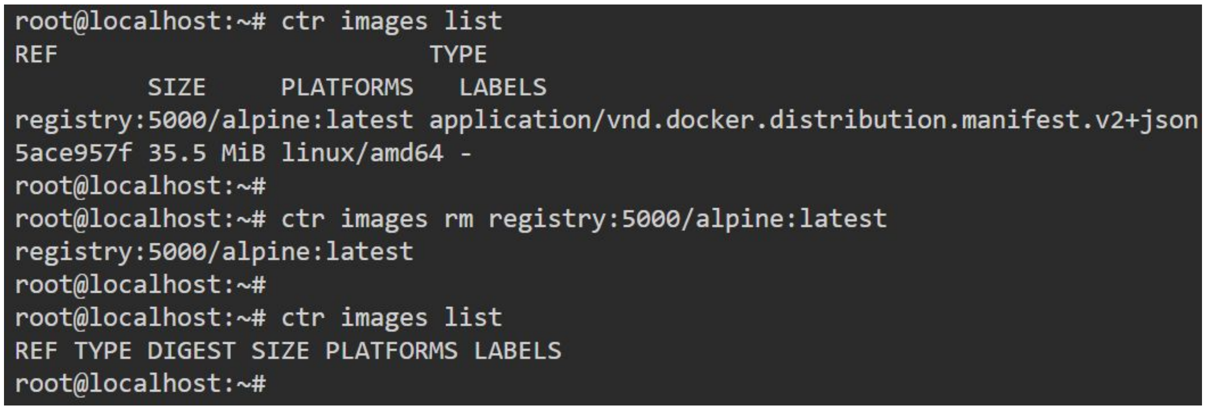
Command: ctr image push --skip-verify --plain-http registry:5000/alpine:latest



```
root@localhost:~# ctr image push --skip-verify --plain-http registry:5000/alpine:latest
manifest-sha256:4963a02ddb4f256659e54b6aeb85303cab638477061bb9978782dd2a5ace957f: done           |++++++++++++++++++++++++++++++++++++++|
config-sha256:84c5dcc06300eaba56eed7a7f94f54cf9ad2658f045fe86c42825296914a9857:   done           |++++++++++++++++++++++++++++++++++++++|
layer-sha256:4dc2274c3171fd1f1957b3fa2c03e2a5d4fadc5171bf03afb0d643392ad9e868:    done           |++++++++++++++++++++++++++++++++++++++|
layer-sha256:89d9c30c1d48bac627e5c6cb0d1ed1eec28e7dbdfbcc04712e4c79c0f83faf17:    done           |++++++++++++++++++++++++++++++++++++++|
elapsed: 0.2 s                                                                    total:   0.0 B (0.0 B/s)

root@localhost:~#
```

16. Remove image

Command: ctr image rm


```
root@localhost:~# ctr images list
REF                         TYPE
         SIZE     PLATFORMS   LABELS
registry:5000/alpine:latest application/vnd.docker.distribution.manifest.v2+json
5ace957f 35.5 MiB linux/amd64 -
root@localhost:~#
root@localhost:~# ctr images rm registry:5000/alpine:latest
registry:5000/alpine:latest
root@localhost:~#
root@localhost:~# ctr images list
REF TYPE DIGEST SIZE PLATFORMS LABELS
root@localhost:~#
```

References:

1. Containerd (https://containerd.io/)