Assignment 3 - Container Complex

Introduction

We have chosen Libcontainer as our container library. Libcontainer provides a native Go implementation for creating containers with namespaces, cgroups, capabilities, and filesystem access controls. It allows you to manage the lifecycle of the container performing additional operations after the container is created.
To run and modify libcontainer we have used runC. RunC is a lightweight universal runtime container. It is a CLI tool for spawning and running containers and is built on libcontainer.

Feature Specification

We have implemented a new state in the container lifecycle i.e. 'Lock' state and ‘Unlock’ state, This feature will prompt the user for a password while locking the container and unlocks the container when given the appropriate password.

Following is a brief description of the states:

1. **Lock**: In this state the container is locked temporarily for some security reasons. If the admin of the container does not want any other unauthorized person to edit the configurations of the container, the container can be put in this state with a password. Upon entering this state the container configuration are frozen and changes cannot be made to it.
2. **Unlock**: This is a state followed by lock state. In this state the container is unlocked after validating the password entered by the user. Since the lock state is password protect we are ensuring here that only the authorized user can put back the container in the unlock state.

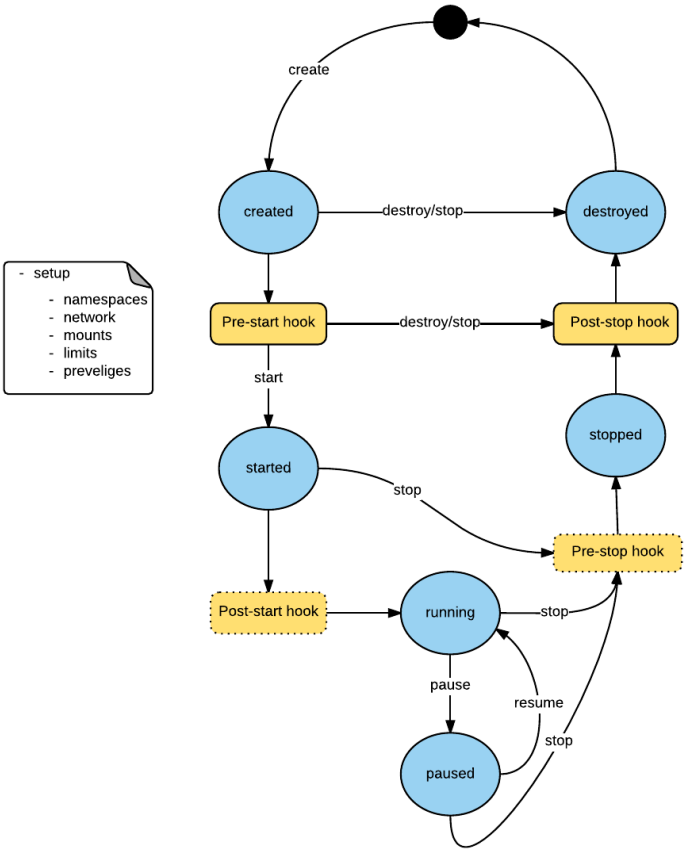
Note: While in the lock state the container cannot be directly destroyed. Since its in the lock state it has to come out of the lock state first and then it can be destroyed, though it can be destroyed in the *paused* state or *running* state.

Example -

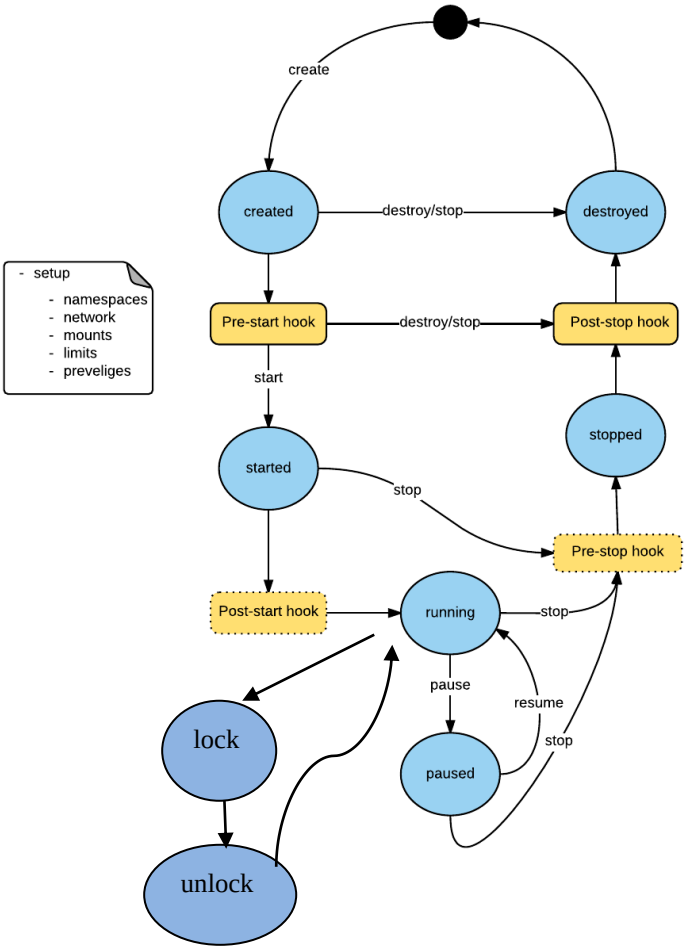
```
#runc lock
Enter password: 123
Re-enter password: 123
Container Locked.

#runc unlock
Enter password: 123
Container Unlocked.
```

The current lifecycle of the container is as follows:



Modified lifecycle:



Design notes

Files modified:

- containers_linux.go
  Contains implementation of container functions and other essential function for running container in linux environment
- container.go
  Contains interface of container functions and state information
- main.go
  starting point of runc api and contains commands for runc
- pause.go
  Contains implementation of lock and unlock command

Functions added/modified:

1. Implementation of *unlock* command
2. Implementation of *lock* command
3. Implementation of *lock* state
4. Implementation of *unlock* state
5. Modified *destroy*() state
6. Gettters and setters for password in container object

Conclusion:

We have provided an extra layer of security to lib container by adding lock and unlock states with password protection and also modified destroy state.