Proceedings of the Second International Conference on Intelligent Computing and Control Systems (ICICCS 2018)
IEEE Xplore Compliant Part Number: CFP18K74-ART; ISBN:978-1-5386-2842-3

# *Build Minimal Docker Container Using Golang*

Biradar Sangam M
Student, Department of CSE,
Alliance University
Bangalore, India.
smbiradar14@gmail.com

Dr.Shekhar R
Department of CSE,
Alliance University
Bangalore, India.
Shekhar.r@alliance.edu.in

Dr. A.Pranayanath Reddy
Department of CSE,
Alliance University
Bangalore, India.
Pranay_taj@yahoo.co.in

*Abstract*— **Docker is a platform which can contain 'N' number of container. Each container internallyis an Encapsulation of Linux kernel feature i.e. namespaces and cgroup. Isolating the process from other process in the same server can be achieved using Namespace isolation. It is done so that it can't see certain portion of overall system. Control groups as its name describes it controls the process, memory and CPU. It also keeps track of used memory. Multiple containers are present on a Docker hub each is having its own virtual environment. It is like a multi tenancy where the Containers are shared through same host kernel but each of it has their own virtualized network adapter and filesystem. It allows to develop, deployment and management applications in efficient manner. Docker containers are developed using Go Language.Go language is used to provision to support system programming to the container.Containers are light weighted and portable,as said it is an encapsulation of an environment in which the application is to be to run. Containers are created from images.Itcontains all dependencies and binaries need for an application to run.Containerization is the new way to build, ship and deploy applications. To create a container with minimal storage is a challenging task. Here in this paper we have tried to show how to create a Docker container with minimal storage.**

***Keywords— Go language,Container,namespaces,cgroups,Docker.***

## I. INTRODUCTION

Go is the compiled and statically typed programming language. It is faster in term of complexity and it also provide many advance feature like concurrency, channel and routines and Go is supportive for environment adopting patterns similar to dynamic language. It is built in the concurrency support: channel, Routine and lightweight processes.

We are using Docker technologies, it is a platform for the developer and sysadmins to build containerized application and its make easy to deployment, shipping to production and run application in different environment. Docker container is made from components so it's easy to the reduce the friction between development, QA and production environment. Soit is ship faster and run the same app to different environment, unchanged, on laptop, data center, VMs and on any cloud.

Docker container is essentially LXC container and they have security feature. When the Docker starts with the docker run command,in background its start with lxc-start to execute continer. Its create some set of namespaces and cgroups for the container that has been created by lxc-start command.

Namespaces are used to providea form of isolation: process running within a container its can't see and even effect, processes running in another container or the host system.[12]

Linux containers provide a virtualized environment for processes on Linux servers with less overhead than virtual machines. System administrators able to deploy containers in less server resource.

Different technologies are collaborated to developing containers possible on Linux, but the concept of namespace isolation is given more importance. Namespaces is Linux feature to allow to server to isolate a process because of that it can't see some portion of overall system . For instance, process ID is used to make process that can only running on server. It has not have ability to send signals to any of processes but in additional feature of container could run its own process id at a init process as a PID 1.and when host system seen the process of container is an entirely different process id.[4]

Linux container provides the capability of implementing all the processes in an isolated fashion by using kernel namespaces and control groups. Each Linux container can view only its file system and process space. Each container is provided an isolated environment with its own computing resources and networking stack. The main advantage of Linux containers is that it doesn't need to run a full Linux OS.

There are two configurations of LXC: Routed-based and bridge-based namespaces which allowcommunicating with outside world. LXC, Docker, Warden are some of the management tools available for Linux Container. [5]

A Linux container is a collection of processes from different containers and which are isolated from the rest of the machine. A container is encapsulation the dependencies to need any application for example, suppose two java application is running on different container and its need different dependency version to run application so container gives the feature of encapsulation. As a result, there is different version of the languages can be in the same environment – without the admin overhead of complete bundles of software or stack its includes the OS kernel and Linux feature. Containerized applications perform about as well as applications deployed on bare metal.

Today's software is very complex so it will be very difficult to maintain , and now days deadline is very important to develop quick software so it will very difficult to IT professional so

following term is useful to improve gap between developer and operation:

- Low cost infrastructure so it will reduce IT staff size
- Quick response to the new businesses
- Keeping data secure
- Adopting modern way of development and hosting services

Linux kernel namespaces are the fundamental building block of containers on Linux. The idea of namespaces as a logical construct is to deal with scope or segmentation, it is a common idea in computer science. Idea of namespaces is used among other interesting concepts such as network or union file systems andmany other computing advancementsother than containers. In Linux, kernel namespaces form a foundational isolation layer that allows for the implementation of Linux containers by creating different user land views.[3]

Control groups: It let you implement mattering and limiting the resources used by processor. In control groups, each subsystem has own hierarchy it like tree with node and each process belong to node.

Control groups is the important key component of the Linux container. Its implement the resources accounting and limiting. Its provides very useful features it also help to ensure that each container get fair of share memory, CPU, disk and more important is its can't put system down to exhaust one of those feature.[12]

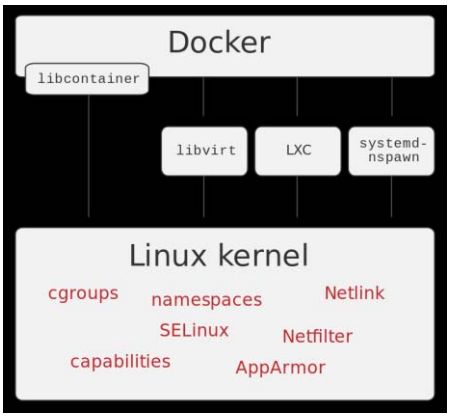


Fig.1. Docker Internal Components

Libcontainer:It is the package in a native GOimplementation for creating namespaces, Cgroups, capabilities and Cgroups access control, It also allows us to manage the lifecycle of the container performing additional operation after the container creation.

Container is produced in a two types of process, for the init stage of execution need binary that execute by container.so for the binaries we are using libcontainer(/proc/self/exc) to be executed as the init process.

## II. METHODOLOGY

### A. *Namespaces*

- IPC NAMESPACES: System V IPC objects and POSIX message queues can utilize their own namespace. CLONE_NEWIPC provides a method for creating objects in an IPC namespace along with other namespaces,These are visible to all other processes that are members of that namespace, but are not visible to processes in other IPC namespaces. This is typically used for shared memory segments. This isolation helps from other IPC related attacks and Denial of Service scenarios.[3]

- MOUNT NAMESPACES: The mount namespace via CLONE_NEWNS is the oldest and only namespace introduced in the 2.4 kernel. The mount namespace provides a process,or group there of treatedas container with a specific view of the system's mounted filesystems. This view can be a mount paths or physical or network drives, or advanced features such as union filesystems, bind mounts, or overlay file systems. The mount namespace can also indirectly secure other namespaces by restricting access to the hosts mounted instance of process, which would violate the PID namespace constraints.[3]

- PID NAMESPACES: It isolate the process ID number space. In other words, processes in different PID namespaces can have the same PID. One of the advantages of PID namespaces is that,when a containers migrates to other hosts it can keep the same process IDs for the processes. PID namespaces allowto each container to have init process that manages various system tasks and its create child process .each process having two PID that is PID inside of the namespaces and outside of namespaces from host system. PID namespaces is nested with process in which PID of each of the layers in hierarchy structure, its starting from the PID namespace in resides of the root PID namespaces. A container will see only process and its send signal kill() from Linux features.it will kill only particular process which given by the PID.[2]

```
root@autb:~# go run main.go run echo sangam
Running [echo sangam]
sangam
root@autb:~# go run main.go run  ps
Running [ps]
  PID TTY          TIME CMD
 2731 pts/1    00:00:00 go
 2744 pts/1    00:00:00 main
 2747 pts/1    00:00:00 bash
 5351 pts/1    00:00:00 bash
11229 pts/1    00:00:00 go
11242 pts/1    00:00:00 main
11246 pts/1    00:00:00 exe
11249 pts/1    00:00:00 ps
```

Fig.2.Displaying PID Instances

- NETWORKING NAMESPACES: each network in its namespace has own network devices, IP addresses, IP routing tables, /proc/net directory, port numbers, and so on.Cgroup is allow limit to given resource and controlling storage, control performance, directory of subsystem performance and isolate the System resource. Networking

feature to each contain that is virtual network device and its own applications that bind to the per-namespace port number space and suitable routing rules in the host system can direct network packets to the network device associated with a specific container. Thus, it is to having the multiple container and its contain web server in same host system each server has hosted on port 80 and every container having own networking namespaces. [2]

- UTS NAMESPACES:(CLONE_NEWUTS)In the context of containers, every container having own hostname and NIS domain for isolating two system . the names can set using the sethostname() and setdomainname() method and its helpful for initialize and configure the script. The term "UTS" derives from the name of the structure passed to the uname() system call: structutsname. The name of that structure in turn derives from "UNIX Time-sharing System".[2]


Fig.3.Display Hostname

- USER NAMESPACE: (CLONE_NEWUSER) isolation is the one of important feature of docker container so in anatomy of container which developed in the golang its use namespace to isolate user and group id number spaces its means user and group id is different inside of namespace and outside of namespaces because of that process can have privilege user id outside a user namespace and inside of namespace user id is zero because of that process will get full root privilege for operation inside the user namespaces and unprivileged process will be outside of namespaces .[2]

### *B. Cgroups*

Control Groups give access control to a process or collection of processes and controlling resource utilization limit. Cgroup and its related subsystem provide the tree like hierarchy structure and its able to inherited and nested to resource control .cgroups isolate and limit a given resource over a collection of processes to control performance or security. A new more powerful and more easily-configured alternative to ulimits/rlimits. To silence the naysayers and doubt over code bloat or added complexity.[3]

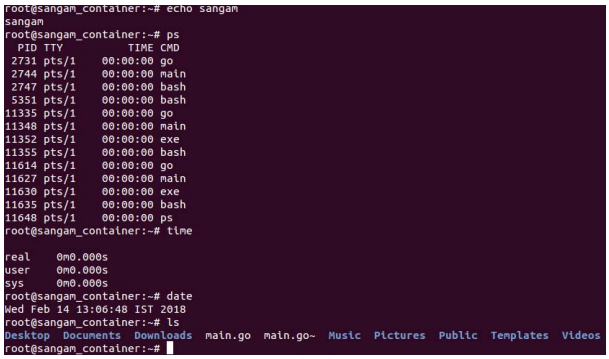


Fig.4. /bin/bash

The Cgroup providing a generic process grouping framework It is a subsystem that does the following along with other features such as Resource Management and Resource Accounting / Tracking solution,

It also handles resources such as memory, CPU, networking and more.

Memory Cgroup: accounting

We will count how much memory used by each process we will track every single memory page.

Memory Cgroups: limits

Each group can have its own limits there are two type of limits soft limits and hard limit.

Limits can set different kind of memory i.e. physical memory, kernel memory and total memory.

### *C. Layered Filesystems*

To isolate resource sharing among different containers we use Namespaces and CGroups. They acts as a big metal sides and the security guard at the dock. The filesystem is used to efficient move all machine images around in basic level layered filesystem its optimize and create copy of filesystem for each container.[1]

### *D. Security responsibility*

Developers can use containers effectively to increase productivity. As they can create a container that contain their application with all dependencies, test them in live environment with all its libraries, and verify whether it works in production environment or not. Operations teams also appreciate containers because they get the applications with in short time and in a cohesive package along with their dependencies and configurations. [4]

## III. IMPLEMENTATION

To construct a container we are using built-in functions and packages of Go language. Following are the packages and methods used:

- fmt
- io/util

- os
- os/exec
- path/filepath
- syscall

fmt- It is a package which is used to implements formatted I/O functions. It is similar to functions printf and scanf in C programming.[7]

io/util: It is a package that implements I/O utility functions.

OS:It is a package that provides a platform-independent interface to operating system functionality. It is same as UNIX however,error handling is same as Go language.When a failing calls it returns values of error rather than error number these feature is not available in syscall package specifically. [6]

OS/EXEC: It is a package that has methods for running external commands(exec). It wraps on OS. Start Process make it easier to remap stdin and stdout, connet I/O with pipes, and do other adjustments.[9]

Unlike the system library call from C and other language, the OS/exec package doesn't contain system shell because of that it will not expand any glob pattern or handler and other expansion , pipelines or redirection by shell . The package behaves more like C's 'exec' family of functions. [9]

Path/filepath: It is a package that is used to implement utility routines, to perform manipulation of filename paths that is compatible for the os, define file path using the filepath package used by the slashes or backslashes and depends on os.[10]

Syscall: It is a package that contains an interface to the low-level operating system primitives. The details depends on underlying operating system. By default Go doc will take syscall of current operating system.[11]

Requirements:

Operating system: FreeBSD 9.3 and later, Linux 2.6 and later, mac os 10.8 and later, windows XP sp2 and later

Tools: VSCODE editor or any other alternative editor

Installation of Golang on Linux:

https://golang.org/dl/the installer as per the operating system

useful command to run the project:

go run < one or more as per program filename >

to run the go program

go run < filename> run bin/bash

to check container is working or not

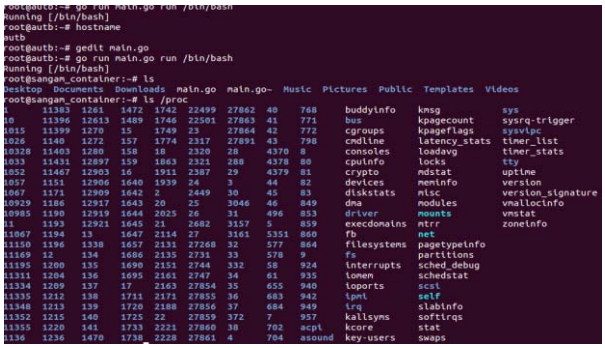


Fig.4 Process list

## IV. CONCUSION

Docker technology is the future of the software development. Container is like package which contain information of all dependencies or needed resources that help in software production. The challenge is creating a container with minimal storage capacity. So, we created minimal container in Go lang. We created a Docker container with minimal storage capacity that can handle deployment process effectively and also shown how they made it from namespaces, Cgroups, filesystem etc. We developed container to run application on own environments. The major advantage of container is build it once and run on any platform and anywhere.

## REFERENCES

[1] Build a container golang -https://www.infoq.com/articles/build-a-container-golang

[2] Michael Kerrisk. "Namespaces in operation, part 1: the namespaces API", January 4, 2013

[3] Aaron Grattafiori,"Understanding and Hardening Linux Containers" June 29, 2016 – Version 1.1

[4] Major Hayden,"Securing Linux Containers GIAC (GCUX) Gold Certification",July 26, 2015

[5] Kotikalapudi sai venkat naresh,"Comparing live migration linux containers and kernel virtual machine", Feb 17

[6] OS package golag - https://golang.org/pkg/os/

[7] Fmt package golang - https://golang.org/pkg/fmt/

[8] IO utiil package - https://golang.org/pkg/io/ioutil/

[9] Exec package golang - https://golang.org/pkg/os/exec/

[10] Filepath package golang- https://golang.org/pkg/path/filepath/

[11] System call package golang- https://golang.org/pkg/syscall/

[12] J. Petazzoni, "Containers & Docker: How Secure Are They?", Aug. 2013, [online] Available: http://blog.docker.com/2013/08/containers-docker-how-secure-are-they.

[13]Creating open standards around container technology. https://github.com/opencontainers/runc/tree/master/libcontainer

[14] Jeff Meyerson,"The Go Programming language",IEEE software, Vol 31, issue 5, sep-oct 2014.