Linux and Tooling Notes

—

Date:

Contents

Chapter 1

Linux

Chapter 2

Bash CLI

First Steps in the Terminal

First Steps with Linux

File Management pt1 - Organised Files and directories

File Management pt2 - Handle text files

Redirection - Mangage Data Streams

Data processing through command chaining

environment variables, manage shell configuration

Create a cuustom bash prompt

How commands are parsed

Chapter 3

Linux

Everything is file concept

Linux user management groups, permissions and access control

Linux orchestrate system operations

Job control in bash

Package management

Package management CentOS

System Boot Processes and SystemMd

Volumes partitions and mounts

logical volume manager

Sfotware upgrade and trouble shooting

Cron jobs, automate and execute tasks

Networking - Arrange and manage linux communication channels

SSH

Setting up a webserver

Control firewall traffic

SELinux - Robust access control

Other linux distributesion

Chapter 4

Bash Scripting

Bash Scripting Basics pt 1

Bash Scripting Basiss pt 2

Numeric Variables and Arithmetic Operations

Access External Data - Retrieve JSON data from APIs

Test and the if clause etc

While loops

For loops

Demo automted thumbnail generation

USer interaction with ”select” etc.....

Arguments in Bash - Process Command LIne Input with Scripts

Functions - Write Well structured scripts

Arrays in Bash - Manage and Manipulate Data Collections

Project - Build a Backup Script with Archiving and Compressions

Effective Data Filtering with grep and Regular expressions

Reliable file fetching

The depths of bash

Customise CLI

Notes on Linux, focus on the stuff beyond the basics.
Everything is a file.....

What is a cgroup

Linux Processes - Orchestration of system operations .....?

Package managers ....meh

System boot process and systemd

Linux processes

Volumes partitions and mounts - Managing file systems

Logical volume manager

SSH hmmmmmmm.....

Firewalls, controlling network traffic

4.1 Useful programs

systemmd systemctl - status — show processes on the linux system

4.2 Notes about the file structure

4.3 Bash

Reminders on Bash related things.....beyond standard programming constructs....

4.3.1 Bash CLI

Useful commands here

4.3.2 Bash Scripting

Hmmmmmm useful stuff for scripting

4.4 Hardening a Linux box

4.4.1 Generally

Hardening a linux box

4.4.2 SELinux

Chapter 5

CI/CD stuff

5.1 Terraform

Motivations Provision AWS infrastructure programmatically.

5.1.1 Intoduction to Infrastrcutre as Code

5.1.2 Getting Started with Terraform

5.1.3 Terraform Basics

5.1.4 Terraform State

5.1.5 Working with Terraform

5.1.6 Terraform with AWS

5.1.7 Remote State

5.1.8 Terraform Provisioners

5.1.9 Terraform Import, Tainting, Resources and Debugging

5.1.10 Terraform Modules

5.1.11 Terraform Funtions and Conditional Expressions

5.2 Terraform: AWS Infrastructure as Code

5.2.1 Introduction to terraform

5.2.2 Terraform Basics

5.2.3 Terraform with AWS

5.2.4 Advanced Terraform Usage

5.2.5 Packer

5.2.6 Docker on AWS using ECS and ECR

5.2.7 Module Development

5.2.8 Advanced Module Development

5.2.9 AWS CodePipeline (Contiuous Delivery / Deployments)

5.2.10 HashiCorp Certificatoin....

5.2.11 Cloud Development Kit for Terraform

5.3 Git

Niche git commands

5.4 Docker

Motivations Solves the problem of setting up environments for applications.

Operating Systems, Hardware and the Kernel Note: Some of these features are Linux specific..... Note: Docker is a Linux virtual machine, inside this Linux VM is where the containers are running. This VM uses the Linux kernel, and it will be doing *Linux* kernel things.

Understanding what a Kernel is. Software process that acts as the middle guy between programs running on a computer and hardware resources. Programs interact with the kernel through system calls. Kernel exposes different system calls. Namespaces..segment hardware resources to different processes, control groups are another term, limit the CPU, RAM etc that a program can use.

Namespacing (Linux specific) - Isolating resources per process (or group of processes) - Processes, Hard Drive, Network, Users, Hostnames, Inter Process Communication

Control groups - Limit amount of processes used per process —- Memory, CPU, HD I/O, Network Bandwith

Container - a process or set or processes that have a grouping of resources specifically assigned to it

Image - A file system snapshot of a very specific set of directories and files. Will also contain a specific startup command.

Kernel will isolate a section of the hardrive/physical resource and make it avaiable only to this container. This container has access to this very specifc segment of the physical resources.

Terminology Docker is an overloaded term. Could be referring to Docker client, docker server, docker hub or docker compose. The goal of these tools is to create and run containers.

Image Single file with all the deps and config required to run a program. Containers are an instance of a program.

Container A program with its own isolated set of hardware resources. Consider kernel.

Docker Client Tool commands are issued to

Docker Server Tool that handles the docker related tasks such as creating images, running containers etc. Docker will check if a local image exists before downloading one from a central repository.

The Docker Ecosystem

- Docker Client - The CLI that interacts with the server
- Docker Server/Daemon - Background Server running on a machine
- Docker Machine - Used to manage docker environments
- Docker Images - The files used to create containers
- Docker Hub - A public repository for storing images
- Docker Compose - Useful for multi container environments, most applications will require different services.

5.5 Docker commands

Docker compose

5.6 Kubernetes

What is Kubernetes?

Clusters - A combination of a master node and multiple nodes. Node - A virtual machine which will run some number of containers, more than likely different. Each virtual machine can run different containers or different images, or different numbers of containers. All these different nodes are managed by the master node. Direction are given to the master node Load balancers are external to the cluster, routing requests.

Use case, running many differnet containers with different images.

Usage

In a development environment use minikube In Prod *managed solutions* are used via Cloud providers. Cloud providers would typically handle the low level details. DIY option does exist.

Interact with minikube using KubeCtl. Manage the nodes with this tool. Only to be used in a local environment.

5.6.1 Docker compose vs Kubernetes

Docker Compose	Kubernetes
Each entry can optionally get docker compose to build an image	Kubernetes expects all images to be alre
Each entry represents a container we want to create	One config file per object we want to c
Each entry defines the networking requirements (ports)	We have to manually set up all netwo

Table 5.1: Sample Table

Kubernetes expects all docker images to be prebuilt. There is no build pipeline in kubernetst. Expectation is during some outside step these things will be built.

5.6.2 Deployment to a Kubernetes cluster

Make sure image is hosted on docker hub Make one config file to create the container Make one config file to create the container

5.7 Jenkins

5.7.1 Getting Started with Jenkins

5.7.2 Jenkins and Docker

5.7.3 Jenkins and AWS

5.7.4 Jenkins and Ansible

5.7.5 Jenkins and Security

5.7.6 Jenkins Tips and Tricks

5.7.7 Jenkins and Email

5.7.8 Jenkins and Maven

5.7.9 Jenkins and Git

5.7.10 Jenkins and DSL

5.7.11 CI/CD Defintions

5.7.12 Jenkins Pipeline - Jenkinsfile

5.7.13 CI/CD + Jenkins Pipeline + Docker + Maven

5.7.14 Manually using Jenkins

Hmmmm.....

Declarative pipeline vs Scripted Pipeline

5.7.15 Jenkins DSL

Create jobs using code. Requires Jenkins DSL plugin and the Groovy programming language.

Seed Jobs The jobs that creates all the other jobs on the server.

5.7.16 Jenkinsfiles

Pipeline The workflow that is executed when going through the CI/CD process (or any other process). Jenkins as code a.k.a pipeline as code. Requires plugin to be installed

Listing 5.1: Jenkinsfile for a CI/CD Pipeline

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                echo 'Building the application...' // Print message
            }
        }

        stage('Test') {
```

```
            steps {
                echo 'Running tests...'
            }
        }

        stage('Deploy') {
            steps {
                echo 'Deploying to production...'
            }
        }
    }

    post {
        success {
            echo 'Pipeline completed successfully!'
        }
        failure {
            echo 'Pipeline failed!'
        }
    }
}
```

5.8 Kubernetes

What it is A Master with one or more Nodes. Nodes are either virtual machines or a physical computer. Each of these machines can be used to run a different set of containers. Can be differnt images or different number of containers. These Nodes are managed by a Master node. Master node has a different set of programs running on it which can controdl what the nodes are running at a given time. The cluster is interacted with through the master.

Why use kubernets - Running different containers with different images....at scal

5.9 Ansible

Ansible for now. Need to consider chef and puppet

5.10 Terraform

Cloud stuff here

Chapter 6

JVM Specific / Delete and or separate

6.1 Maven

Maven stuff or keep separated

6.2 JVM details

Maybe include some low level JVM stuff here....or keep separate

Chapter 7

Hmm maybe more, maybe delete

Other tools that can be used via a shell

Code Example

7.1 Code Example

Listing 7.1: Java Code for a Simple Cache

```
public class SimpleCache {
    private Map<String, String> cache = new HashMap<>();

    public String get(String key) {
        return cache.get(key);
    }

    public void put(String key, String value) {
        cache.put(key, value);
    }
}
```