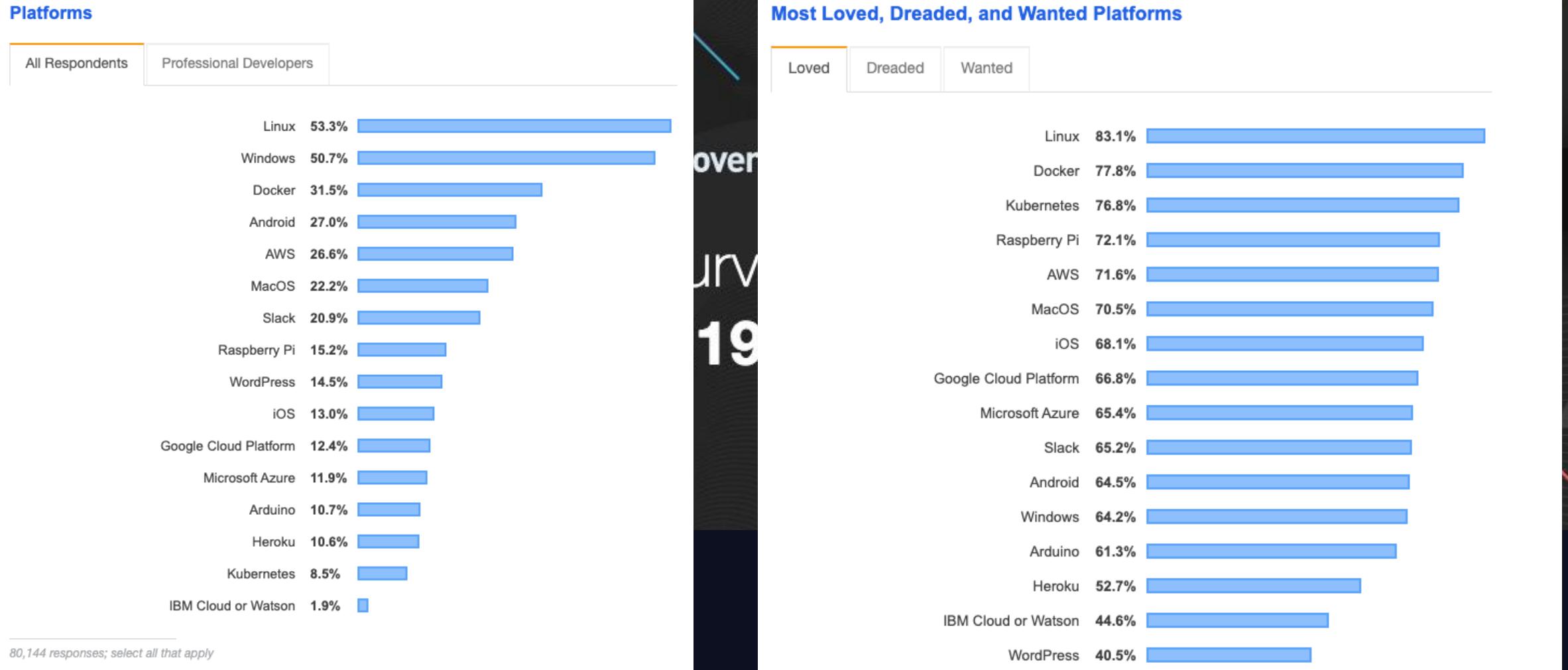




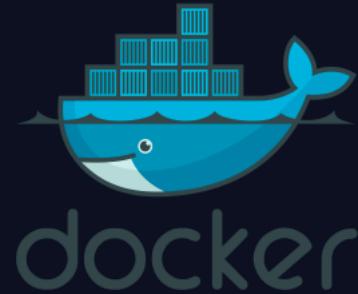
KodeKloud

Learn more about DevOps and Cloud courses with KodeKloud: <https://kode.wiki/3N3A4kt>

Why Linux?



Why Linux?



2013 - Docker was born
2016 – Docker for Windows was born



Can Ansible run on Windows? ANSIBLE DOCUMENTATION

No, Ansible can only manage Windows hosts. Ansible cannot run on a Windows host natively, though it can run under the Windows Subsystem for Linux (WSL).

https://docs.ansible.com/ansible/latest/user_guide/windows_faq.html



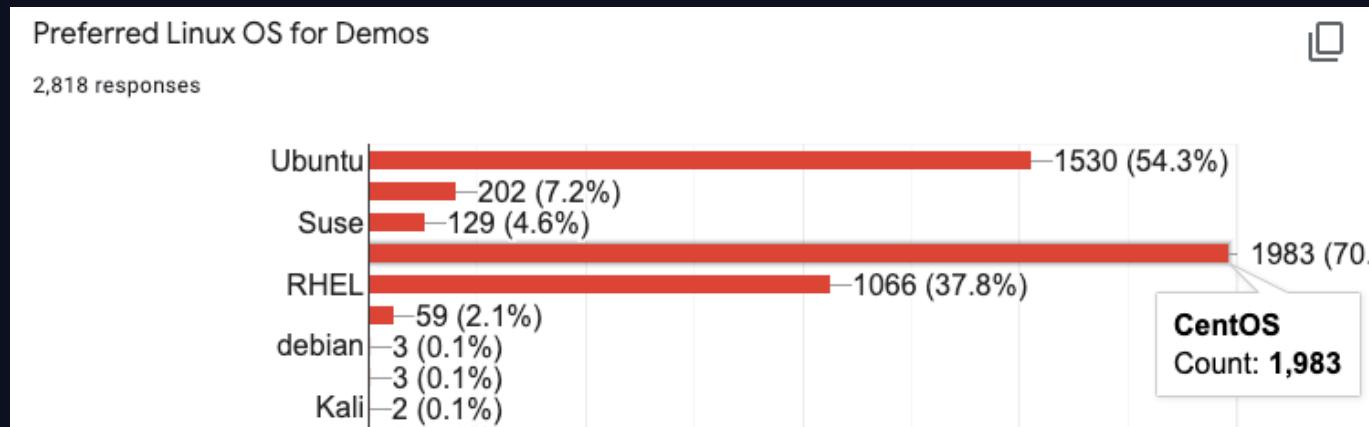
Note: The Kubernetes control plane, including the [master components](#), continues to run on Linux. There are no plans to have a Windows-only Kubernetes cluster.

[Kubernetes Documentation](#)

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

Linux Basics

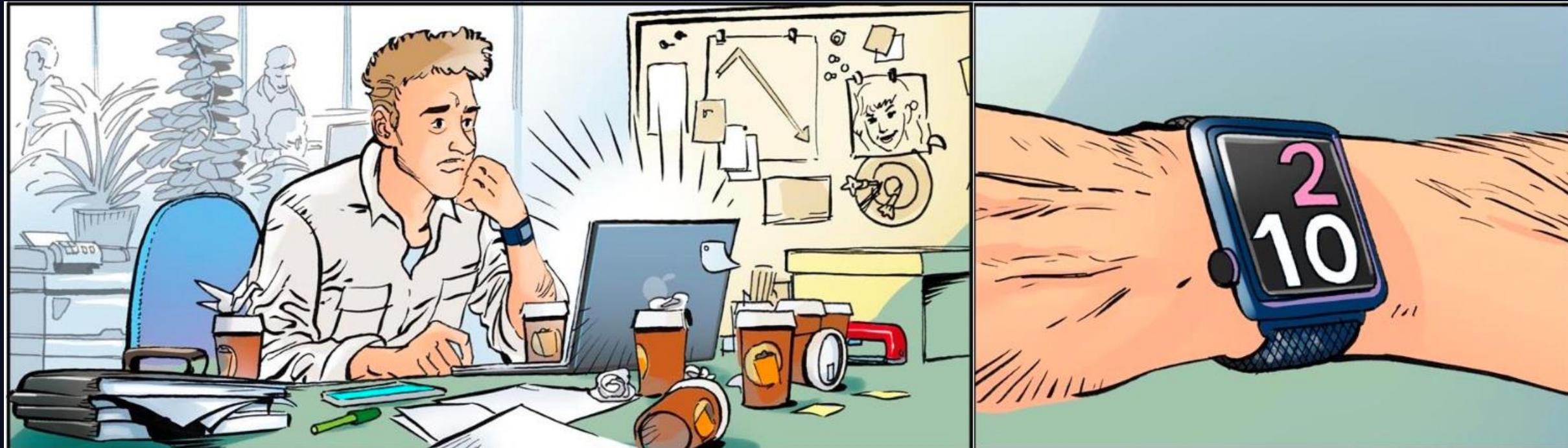
- Linux CLI
- VI Editor
- Package Management
- Service Management



*Allowed multiple selections

Linux Basics Course

www.kodekloud.com



just enough



LINUX

Shell Types

```
▶ echo $SHELL  
/bin/bash
```

Bourne Shell (Sh Shell)

C Shell (csh or tcsh)

Z Shell (zsh)

Bourne again Shell (bash)

Basic Commands

```
▶ echo Hi
```

```
Hi
```

Print to screen

```
▶ ls
```

```
File.txt my_dir1 file2.conf
```

List files & folders

```
▶ cd my_dir1
```

Change directory

```
▶ pwd
```

```
/home/my_dir1
```

Present Working Directory

```
▶ mkdir new_directory
```

Make Directory

```
▶ cd new_directory; mkdir www; pwd
```

```
© /home/my_dir1/new_directory
```

Multiple commands

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

Commands - Directories

/tmp/asia/india/bangalore

```
▶ mkdir /tmp/asia  
▶ mkdir /tmp/asia/india  
▶ mkdir /tmp/asia/india/bangalore
```



```
▶ mkdir -p /tmp/asia/india/bangalore
```

Make Directory Hierarchy

```
▶ rm -r /tmp/my_dir1
```

Remove Directory

```
▶ cp -r my_dir1 /tmp/my_dir1
```

Copy Directory

Commands - Files

```
▶ touch new_file.txt
```

Hi

Create a new file (no contents)

```
▶ cat > new_file.txt
```

This is some sample contents

CTRL + D

Add contents to file

```
▶ cat new_file.txt
```

This is some sample contents

View contents of file

```
▶ cp new_file.txt copy_file.txt
```

Copy File

```
▶ mv new_file.txt sample_file.txt
```

Move (Rename) File

```
▶ rm new_file.txt
```

Remove (Delete) File

Labs

- NOTE: These labs should NOT have the user require to use sudo
- Have a file and folder tree structure created. Ask MCQ questions (only based on the previous slides) to users, such as
 - Identify number of files and directories in a path
 - What is a file not present in a directory
 - Navigate to some directory(don't specify the full path) and ask user to find the present working directory
- Ask user to perform operations
 - Create empty file
 - Create file with some content
 - Create directory
 - Create directory hierarchy
 - Copy file to a different place (Stage a file with some data)
 - Copy directory
 - Remove file
 - Remove directory and all contents



KodeKloud

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

just a bit more



LINUX

User Accounts

```
▶ whoami
```

```
matthew
```



matthew

```
▶ id
```

```
uid=1001(matthew) gid=1001(matthew) groups=1001(matthew)
```

```
▶ su aparna
```

```
Password:
```

```
▶ ssh aparna@192.168.1.2
```

User Accounts

```
▶ ls /root
```

```
ls: cannot open directory /root: Permission denied
```

```
▶ sudo ls /root
```

```
anaconda-ks.cfg initial-setup-ks.cfg
```



matthew



root

SUDO
/etc/sudoers

Download Files

```
▶ curl http://www.some-site.com/some-file.txt -O  
some-file.txt
```

```
▶ wget http://www.some-site.com/some-file.txt -O some-file.txt  
some-file.txt
```

Check OS Version

```
▶ ls /etc/*release*
```

```
/etc/centos-release      /etc/os-release   /etc/system-release  
/etc/centos-release-upstream /etc/redhat-release /etc/system-release-cpe
```

```
▶ cat /etc/*release*
```

```
CentOS Linux release 7.7.1908 (Core)  
Derived from Red Hat Enterprise Linux 7.7 (Source)  
NAME="CentOS Linux"  
VERSION="7 (Core)"  
ID="centos"  
ID_LIKE="rhel fedora"  
VERSION_ID="7"  
PRETTY_NAME="CentOS Linux 7 (Core)"  
ANSI_COLOR="0;31"  
CPE_NAME="cpe:/o:centos:centos:7"  
HOME_URL="https://www.centos.org/"  
BUG_REPORT_URL="https://bugs.centos.org/"
```

- Wget
- Curl
- Cat /etc/*release*



KodeKloud

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

Package Managers

RPM (Red Hat Package Manager)

► rpm -i telnet.rpm

Install Package

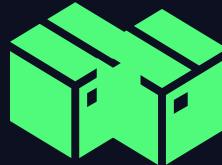
► rpm -e telnet.rpm

Uninstall Package

► rpm -q telnet.rpm

Query Package

?
?
?
?



telnet.rpm

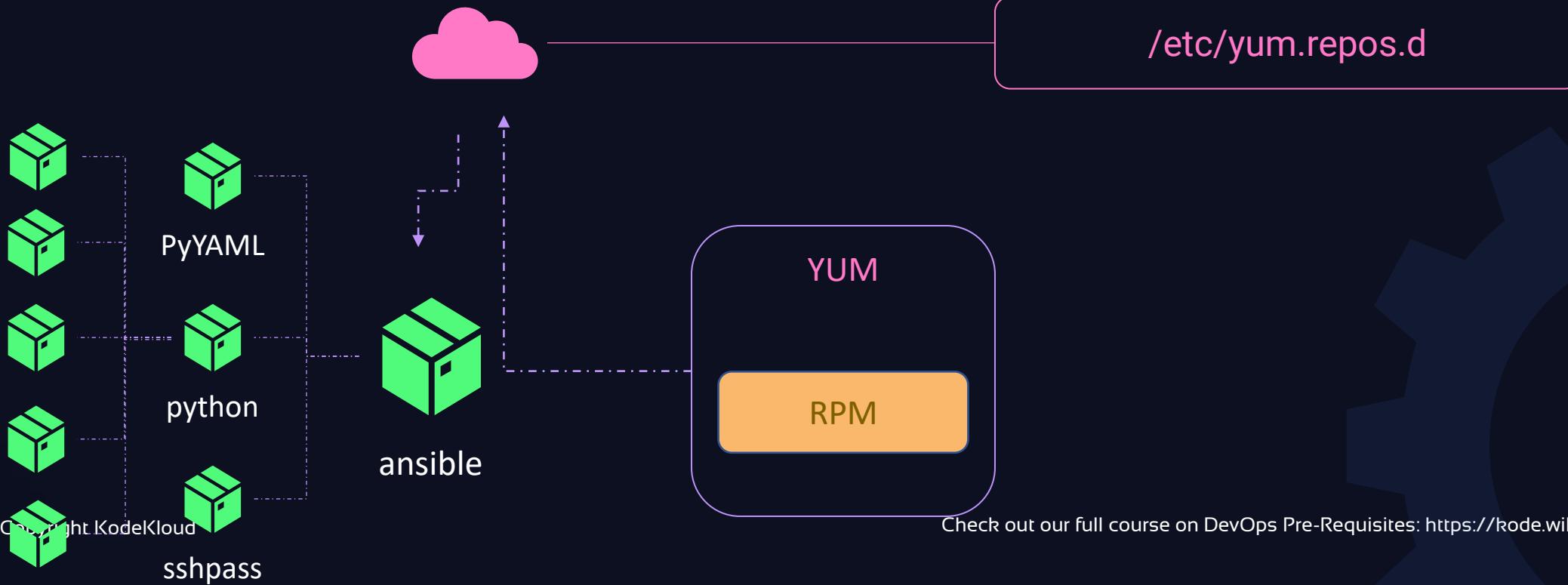
RPM

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

YUM

▶ `yum install ansible`

Install Package



YUM Repos

```
▶ yum
repo id: base/7
extras/
mongo
mysql-
mysql-
mysql8
update
```

```
▶ ls
CentOS
CentOS
CentOS
CentOS
```

```
▶ cat
[extras]
name=O
baseurl=
```

[🔗 Extra Packages for Enterprise Linux \(EPEL\)](#)

Welcome to the home of the EPEL Special Interest Group.

[🔗 Quickstart](#)

- [epel-release-latest-6](#)
- [epel-release-latest-7](#)
- [epel-release-latest-8](#)

You may retrieve signed binary configuration files from one the above two links (varying by the major release number) automatically installed by root thus:

- RHEL/CentOS 6:

```
# yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
```

- RHEL/CentOS 7:

```
# yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

YUM

```
▶ yum list ansible
```

Installed Packages

ansible.noarch	2.9.6-1.el7	@epel
----------------	-------------	-------

```
▶ yum remove ansible
```

```
▶ yum --showduplicates list ansible
```

Available Packages

ansible.noarch	2.4.2.0-2.el7	extras
ansible.noarch	2.9.6-1.el7	epel

```
▶ yum install ansible-2.4.2.0
```

Labs

- View installed packages
- Identify versions of installed packages
- Install specific packages using yum
- Install specific packages using yum with specific versions
- Remove packages



KodeKloud

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

Services

Services

```
▶ service httpd start
```

Or

```
▶ systemctl start httpd
```

Start HTTPD service

```
▶ systemctl stop httpd
```

Stop HTTPD service

```
▶ systemctl status httpd
```

Check HTTPD service Status

```
▶ systemctl enable httpd
```

Configure HTTPD to start at startup

```
▶ systemctl disable httpd
```

Configure HTTPD to not start at startup

Services

▶ /usr/bin/python3 /opt/code/my_app.py

```
* Serving Flask app "my_app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

▶ systemctl start my_app

▶ systemctl stop my_app

/etc/systemd/system

▶ curl http://localhost:5000

```
Hello, World!
```

Services

/etc/systemd/system

▶ /usr/bin/python3.8 opt/prototype/app/app.py

```
* Serving Flask app "my_app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

my_app.service

[Service]

ExecStart=

▶ curl http://localhost:5000

Hello, World!

▶ systemctl daemon-reload

▶ systemctl start my_app

Services

/etc/systemd/system

▶ `systemctl status my_app`

```
● my_app.service
  Loaded: loaded (/etc/systemd/system/my_app.service; static; vendor preset: disabled)
  Active: active (running) since Tue 2020-04-07 09:01:39 UTC; 2s ago
    Main PID: 5038 (python3)
   CGroup: /system.slice/my_app.service
           └─5038 /usr/bin/python3 /tmp/app/my_app.py
```

```
Apr 07 09:01:39 systemd[1]: Started my_app.service.
Apr 07 09:01:39 python3[5038]: * Serving Flask app "my_app" (lazy loading)
Apr 07 09:01:39 python3[5038]: * Environment: production
Apr 07 09:01:39 python3[5038]: WARNING: This is a development server. Do not use it in a produ...ent.
Apr 07 09:01:39 python3[5038]: Use a production WSGI server instead.
Apr 07 09:01:39 python3[5038]: * Debug mode: off
Apr 07 09:01:39 python3[5038]: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
Hint: Some lines were ellipsized, use -l to show in full.
```

my_app.service

[Service]

`ExecStart= /usr/bin/python3 /opt/code/my_app.py`

▶ `systemctl daemon-reload`

▶ `systemctl start my_app`

▶ `systemctl stop my_app`

▶ `curl http://localhost:5000`

Hello, World!

© Copyright Kodekloud

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

Services

/etc/systemd/system

▶ `systemctl status my_app`

```
● my_app.service
  Loaded: loaded (/etc/systemd/system/my_app.service; static; vendor preset: disabled)
  Active: active (running) since Tue 2020-04-07 09:01:39 UTC; 2s ago
    Main PID: 5038 (python3)
   CGroup: /system.slice/my_app.service
          └─5038 /usr/bin/python3 /tmp/app/my_app.py
```

```
Apr 07 09:01:39 systemd[1]: Started my_app.service.
Apr 07 09:01:39 python3[5038]: * Serving Flask app "my_app" (lazy loading)
Apr 07 09:01:39 python3[5038]: * Environment: production
Apr 07 09:01:39 python3[5038]: WARNING: This is a development server. Do not use it in a produ...ent.
Apr 07 09:01:39 python3[5038]: Use a production WSGI server instead.
Apr 07 09:01:39 python3[5038]: * Debug mode: off
Apr 07 09:01:39 python3[5038]: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
Hint: Some lines were ellipsized, use -l to show in full.
```

my_app.service

[Service]

`ExecStart= /usr/bin/python3 /opt/code/my_app.py`

[Install]

`WantedBy=multi-user.target`

▶ `systemctl daemon-reload`

▶ `systemctl start my_app`

▶ `systemctl stop my_app`

Check `systemctl enable my_app`

▶ `curl http://localhost:5000`

Hello, World!

© Copyright KodeCloud

[View on GitHub](#) [View on Wiki](#)

Services

/etc/systemd/system

▶ `systemctl status my_app`

```
● my_app.service
  Loaded: loaded (/etc/systemd/system/my_app.service; static; vendor preset: disabled)
  Active: active (running) since Tue 2020-04-07 09:01:39 UTC; 2s ago
    Main PID: 5038 (python3)
   CGroup: /system.slice/my_app.service
           └─5038 /usr/bin/python3 /tmp/app/my_app.py
```

```
Apr 07 09:01:39 systemd[1]: Started my_app.service.
Apr 07 09:01:39 python3[5038]: * Serving Flask app "my_app" (lazy loading)
Apr 07 09:01:39 python3[5038]: * Environment: production
Apr 07 09:01:39 python3[5038]: WARNING: This is a development server. Do not use it in a produ...ent.
Apr 07 09:01:39 python3[5038]: Use a production WSGI server instead.
Apr 07 09:01:39 python3[5038]: * Debug mode: off
Apr 07 09:01:39 python3[5038]: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
Hint: Some lines were ellipsized, use -l to show in full.
```

my_app.service

[Unit]

Description=My python web application

[Service]

ExecStart=/usr/bin/python3 /opt/code/my_app.py

ExecStartPre=/opt/code/configure_db.sh

ExecStartPost=/opt/code/email_status.sh

[Install]

WantedBy=multi-user.target

▶ `curl http://localhost:5000`

Hello, World!

© Copyright Kodekloud

▶ `systemctl daemon-reload`

Check for updates → Full → Distro → Reload → https://www.fedoraproject.org/wiki/43z8frg

▶ `systemctl start my_app`

Services

/etc/systemd/system

► `systemctl status my_app`

- my_app.service
 Loaded: loaded (/etc/systemd/system/my_app.service; static; vendor preset: disabled)
 Active: active (running) since Tue 2020-04-07 09:01:39 UTC; 2s ago
 Main PID: 5038 (python3)
 CGroup: /system.slice/my_app.service
 └─5038 /usr/bin/python3 /tmp/app/my_app.py

```
Apr 07 09:01:39 systemd[1]: Started my_app.service.
Apr 07 09:01:39 python3[5038]: * Serving Flask app "my_app" (lazy loading)
Apr 07 09:01:39 python3[5038]: * Environment: production
Apr 07 09:01:39 python3[5038]: WARNING: This is a development server. Do not use it in a produ...ent
Apr 07 09:01:39 python3[5038]: Use a production WSGI server instead.
Apr 07 09:01:39 python3[5038]: * Debug mode: off
Apr 07 09:01:39 python3[5038]: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
Hint: Some lines were ellipsized, use -l to show in full.
```

curl http://localhost:5000

Hello, World!

© Copyright KodeKloud

my_app.service

[Unit]

Description=My python web application

[Service]

ExecStart=/usr/bin/python3 /opt/code/my_app.py

ExecStartPre=/opt/code/configure_db.sh

ExecStartPost=/opt/code/email_status.sh

Restart=always

[Install]

WantedBy=multi-user.target

► `systemctl daemon-reload`

Check: [Solve](#) | [P](#) | [C](#) | [P](#) | [P](#) | [Edit](#) | [Help](#) | [About](#) | [Wiki](#) | [43z8frq](#)

▶ `systemctl start my_app`

Service Unit File - Docker

```
/lib/systemd/system/docker.service
```

```
[Unit]
Description=Docker Application Container Engine
Documentation=https://docs.docker.com
BindsTo=containerd.service
After=network-online.target firewalld.service containerd.service
Wants=network-online.target
Requires=docker.socket

[Service]
Type=notify
ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
ExecReload=/bin/kill -s HUP $MAINPID
Restart=always
StartLimitBurst=3
StartLimitInterval=60s
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity

[Install]
WantedBy=multi-user.target
```



KodeKloud

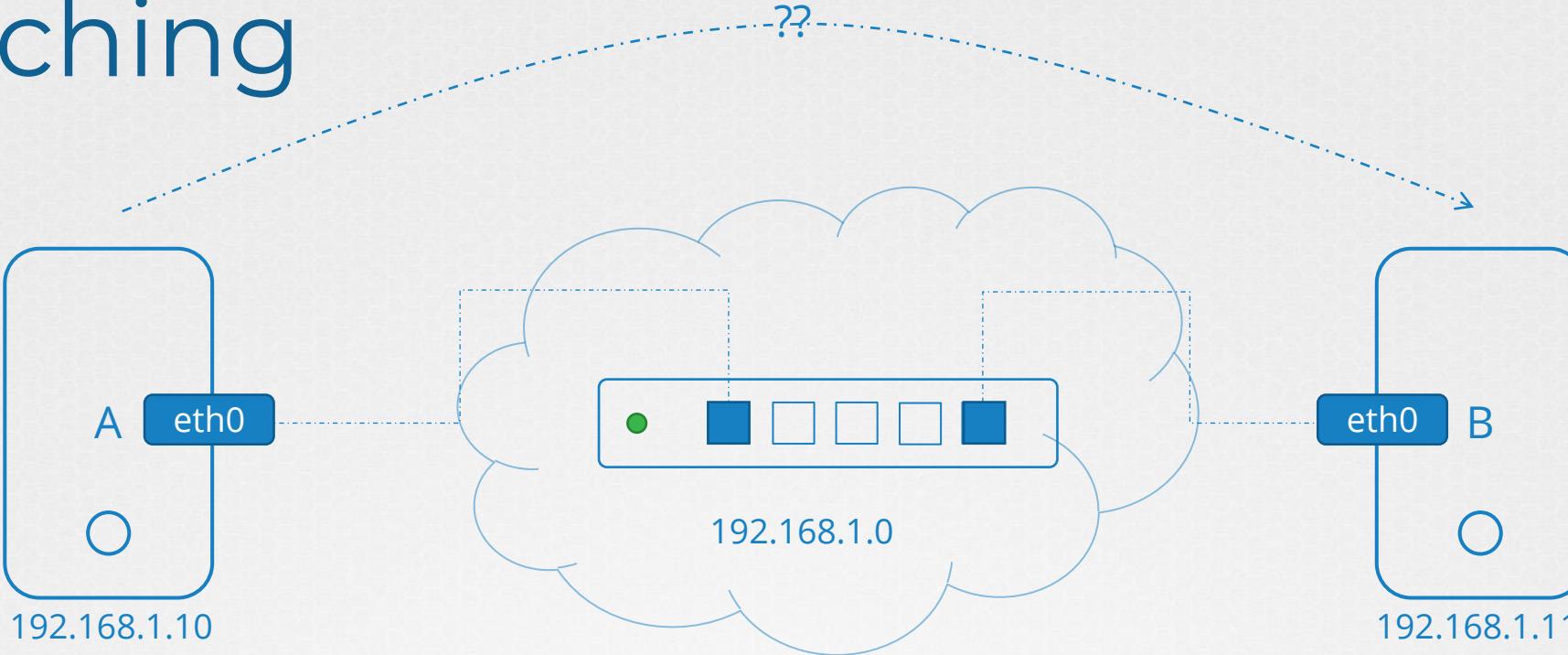
Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

Networking Basics

Networking Pre-Requisite

- Switching
- Routing
- Default Gateway
- DNS Configuration on Linux

Switching



```
▶ ip link
```

```
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc  
fq_codel state UP mode DEFAULT group default qlen 1000
```

```
▶ ip link
```

```
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc  
fq_codel state UP mode DEFAULT group default qlen 1000
```

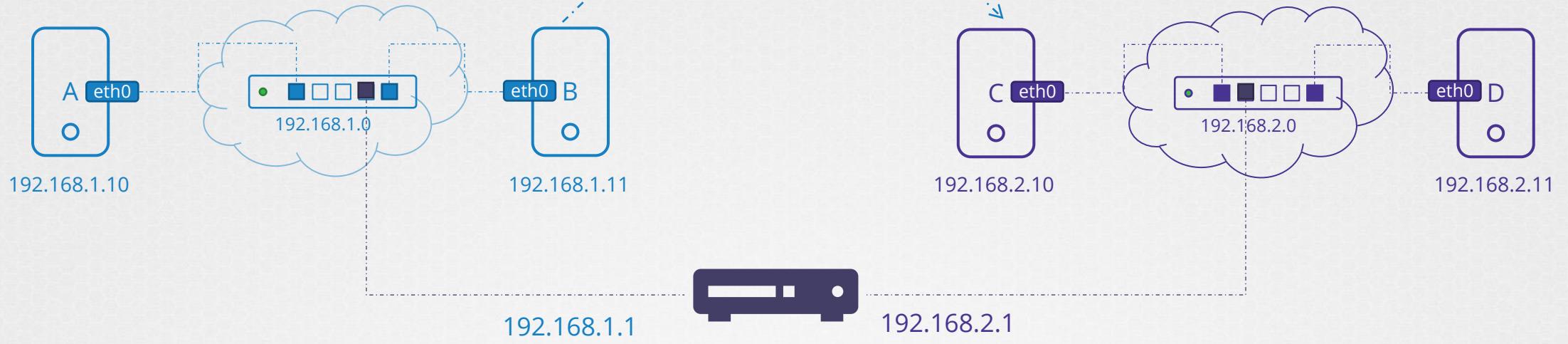
```
▶ ip addr add 192.168.1.10/24 dev eth0
```

```
▶ ip addr add 192.168.1.11/24 dev eth0
```

```
▶ ping 192.168.1.11
```

```
Reply from 192.168.1.11: bytes=32 time=4ms TTL=117  
Reply from 192.168.1.11: bytes=32 time=4ms TTL=117
```

Routing



Gateway



▶ route

```
Kernel IP routing table
Destination      Gateway          Genmask        Flags Metric Ref  Use Iface
```

▶ ip route add 192.168.2.0/24 via 192.168.1.1

▶ route

```
Kernel IP routing table
Destination      Gateway          Genmask        Flags Metric Ref  Use Iface
192.168.2.0      192.168.1.1    255.255.255.0  UG     0       0       0 eth0
```

Gateway

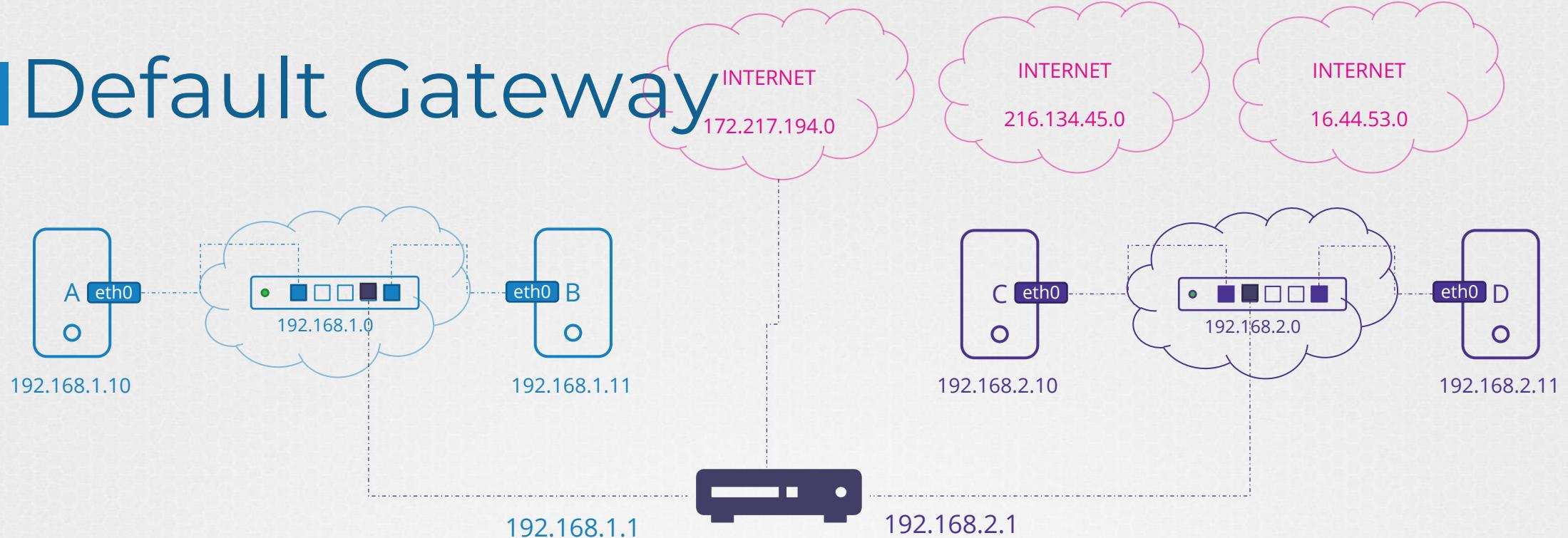


```
▶ ip route add 192.168.1.0/24 via 192.168.2.1
```

```
▶ route
```

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.1.0	192.168.2.1	255.255.255.0	UG	0	0	0	eth0

Default Gateway



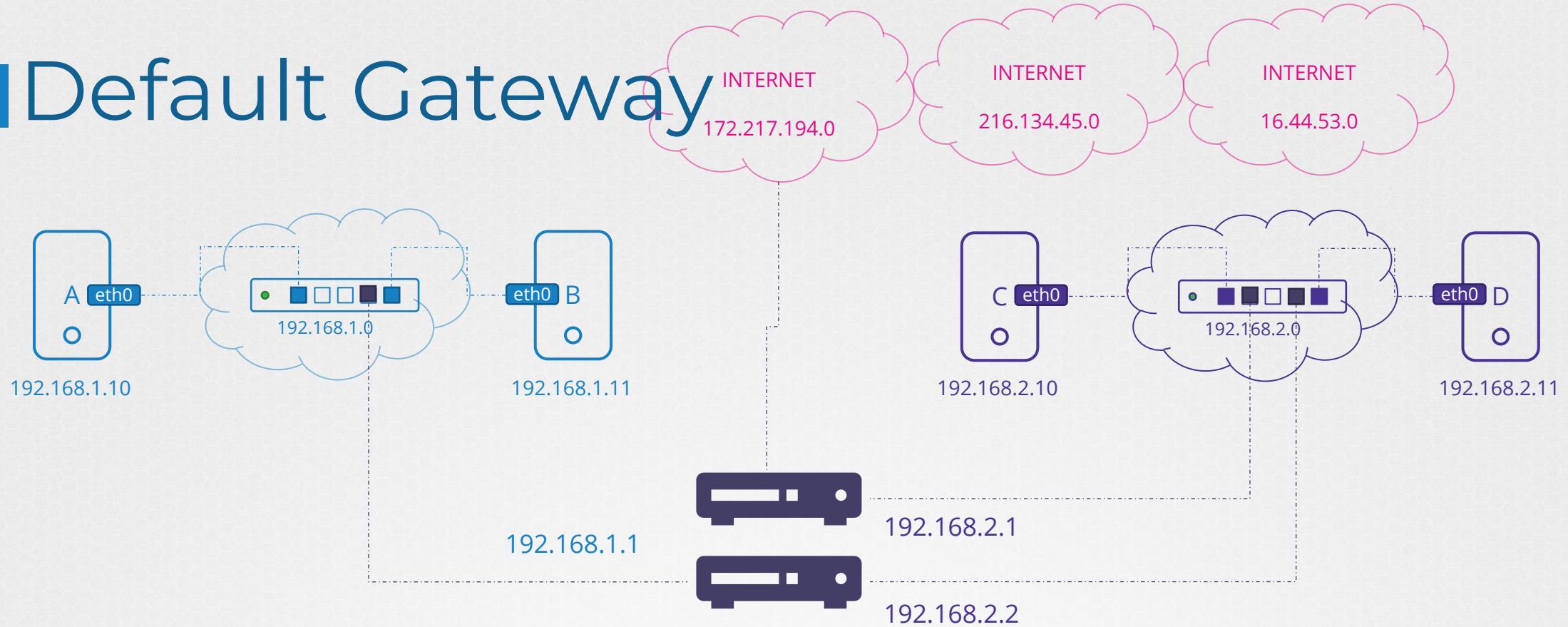
```
▶ ip route add 192.168.1.0/24 via 192.168.2.1
```

```
▶ ip route add default via 192.168.2.1
```

```
▶ route
```

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.1.0	192.168.2.1	255.255.255.0	UG	0	0	0	eth0
0.0.0.0	192.168.2.1	255.255.255.0	UG	0	0	0	eth0
192.168.2.0	0.0.0.0	255.255.255.0	UG	0	0	0	eth0

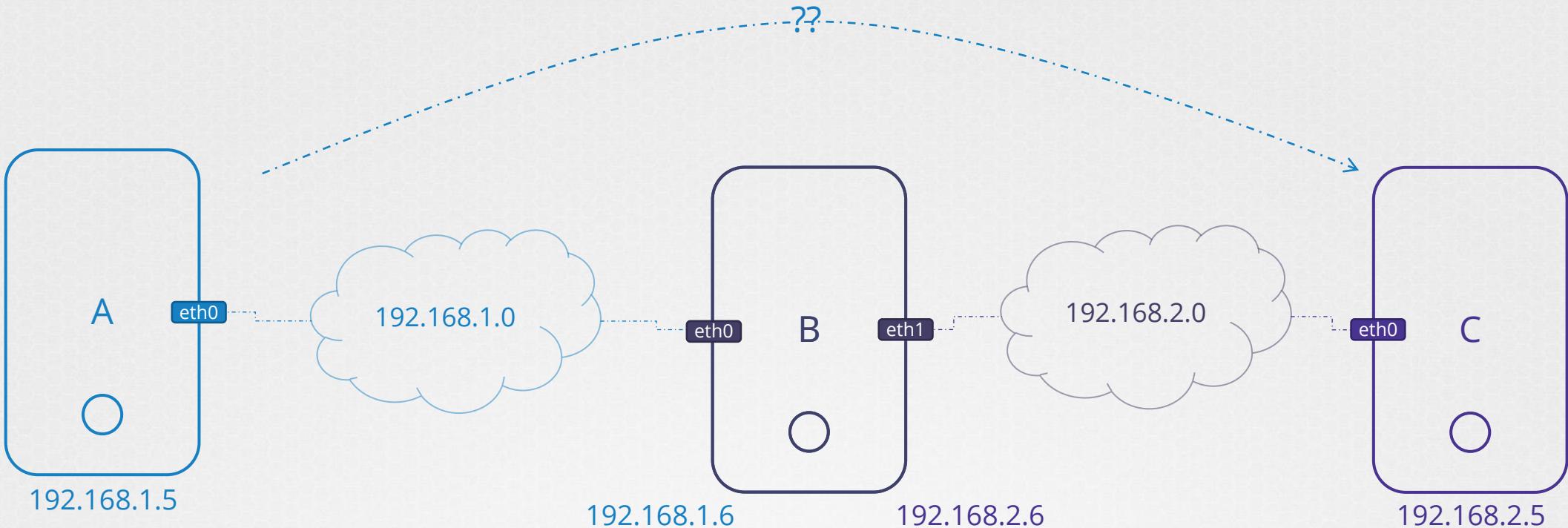
Default Gateway



```
▶ ip route add 192.168.1.0/24 via 192.168.2.2
```

```
▶ route
```

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	192.168.2.1	255.255.255.0	UG	0	0	0	eth0
192.168.1.0	192.168.2.2	255.255.255.0	UG	0	0	0	eth0



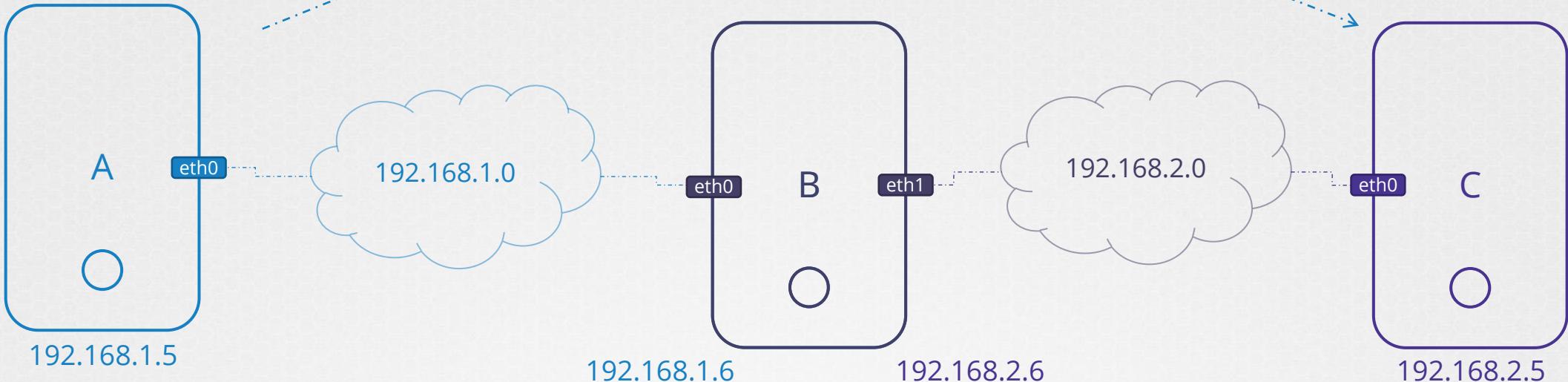
```
▶ ping 192.168.2.5
```

```
Connect: Network is unreachable
```

```
▶ ip route add 192.168.2.0/24 via 192.168.1.6
```

```
▶ ip route add 192.168.1.0/24 via 192.168.2.6
```

```
▶ ping 192.168.2.5
```



```
▶ cat /proc/sys/net/ipv4/ip_forward
```

```
0
```

```
/etc/sysctl.conf
```

```
...  
net.ipv4.ip_forward = 1  
...
```

```
▶ echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
1
```

```
▶ ping 192.168.2.5
```

```
Reply from 192.168.2.5: bytes=32 time=4ms TTL=117  
Reply from 192.168.2.5: bytes=32 time=4ms TTL=117  
Reply from 192.168.2.5: bytes=32 time=4ms TTL=117  
Reply from 192.168.2.5: bytes=32 time=4ms TTL=117
```

Take Aways

```
▶ ip link
```

```
▶ ip addr
```

```
▶ ip addr add 192.168.1.10/24 dev eth0
```

```
▶ ip route
```

```
▶ route
```

```
▶ ip route add 192.168.1.0/24 via 192.168.2.1
```

```
▶ cat /proc/sys/net/ipv4/ip_forward
```

```
1
```



KodeKloud

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

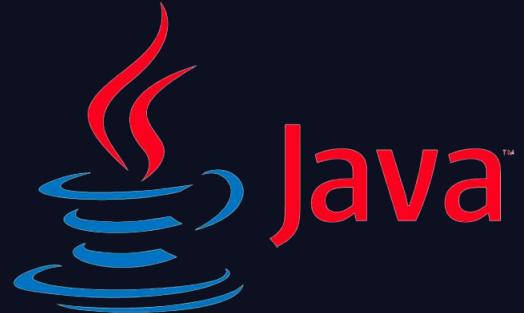
just enough



Introduction

About Java

- Free
- Open-source
- Huge Community



Version	Date
13	2019
12	2019
11	2018
10	2018
9	2017
8	2014
7	2011
6	2006
5	2004

Install Java

```
▶ wget https://download.java.net....  
openjdk-13.0.2_linux-x64_bin.tar.gz
```

```
▶ tar -xvf openjdk-13.0.2_linux-x64_bin.tar.gz  
/opt/jdk-13/bin/java -version
```

```
▶ jdk-13.0.2/bin/java -version  
openjdk version "13.0.2" 2020-01-14  
OpenJDK Runtime Environment (build 13.0.2+8)  
OpenJDK 64-Bit Server VM (build 13.0.2+8, mixed mode, sharing)
```

```
▶ java -version  
openjdk version "1.8.0_242"  
OpenJDK Runtime Environment (build 1.8.0_242-b08)  
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)
```

The screenshot shows the official Java Development Kit (JDK) website at <https://jdk.java.net/13/>. The main content includes:

- GA Releases:** JDK 13, JMC 7
- Early-Access Releases:** JDK 15, JDK 14, JDK 14 macOS Catalina, Jpackage, Loom, OpenFX, Panama, Valhalla
- Reference Implementations:** Java SE 14, Java SE 13, Java SE 12, Java SE 11, Java SE 10, Java SE 9, Java SE 8, Java SE 7
- Feedback:** Report a bug, Archive
- Documentation:** Release notes, API Javadoc
- Build 8 (2019/12/11): General Availability**
 - Changes in this build
 - Issues addressed
- Version** **Name** table:

Version	Name
13	13
12	12
11	11
10	10
9	9
8	1.8
7	1.7
6	1.6
5	1.5

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>
<https://jdk.java.net/13/>



Java Development Kit (JDK)

Develop



jdb



javadoc

Build



javac



jar

Run



JRE

(Java Runtime Environment)



java

```
▶ ls jdk-13.0.2/bin
```

```
jaotc  javadoc  jdeprscan  jinfo  jps      jstaddr  rmiregistry  
jar    javap    jdeps     jjs   jrungscript  keytool  serialver  
jarsigner  jcmand  jfr     jlink  jshell    pack200  unpack200  
java   jconsole  jhsdb    jmap   jstack    rmic  
javac  jdb     jimage.  jmod   jstat    rmid
```

Before v9

Java Development Kit (JDK)

Develop

Build

Run

jdb

javac

javadoc

jar

java



JRE

(Java Runtime Environment)

Download JDK

.tar.gz - 106 MB

[Checksum](#)

Download JRE

.tar.gz - 39 MB

[Checksum](#)

Check o

After v9



Java Development Kit (JDK)

Develop



jdb



javadoc

Build



javac



jar

Run



JRE

(Java Runtime Environment)



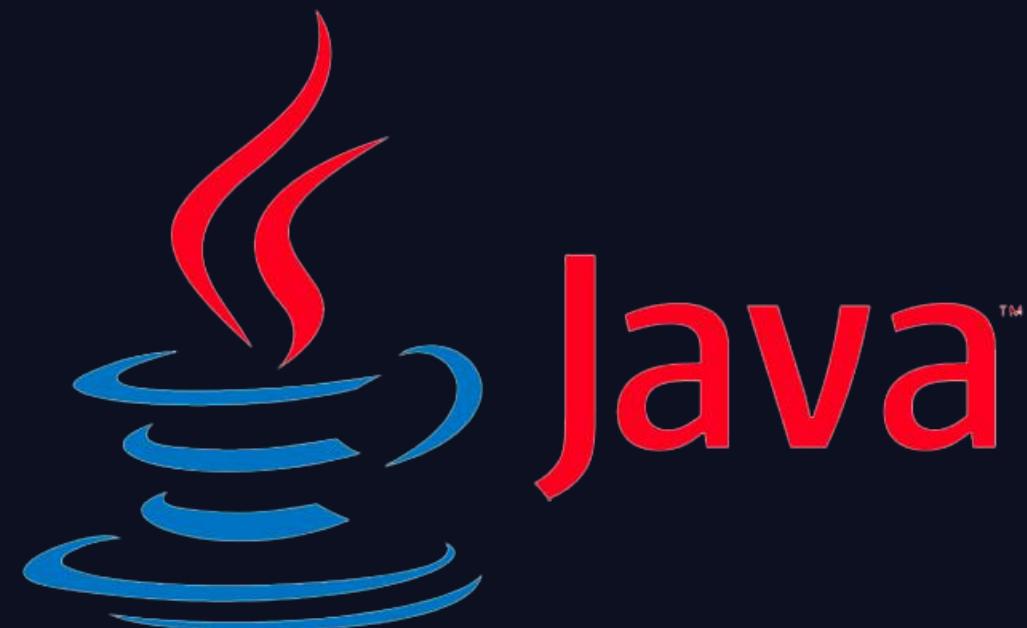
java



KodeKloud

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

just enough



Build

Compile

1. Develop Source Code

MyClass.java

```
public class MyClass {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

2. Compile

► javac MyClass.java

MyClass.class

3. Run

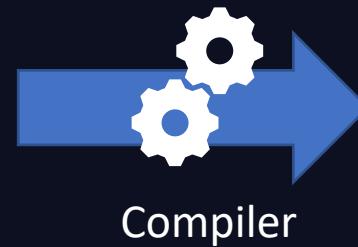
► java MyClass

Hello World

Compile

MyClass.java

```
public class MyClass {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```



Machine Code

```
01101000 10111100 10000001  
01100100 01011100 00010111  
00001010 00001110 11111010  
10110001 01101000 10111100  
10000001 01100100 01011100  
00010111 00001010 00001110  
11111010 10110001 10110001
```

Human Readable
Source Code

Machine Readable
Machine Code

Java Virtual Machine

MyClass.java

```
public class MyClass {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```



Compiler

MyClass.class

```
0:  iconst_2  
1:  istore_1  
2:  iload_1  
3:  sipush 1000  
6:  if_icmpge 44  
9:  iconst_2  
10:  istore_2  
11:  iload_2  
12:  iload_1  
13:  if_icmpge 31
```

Human Readable
Source Code

Intermediary
Byte Code

JVM

Machine Code

```
01101000 10111100 10000001  
01100100 01011100 00010111  
00001010 00001110 11111010  
10110001 01101000 10111100  
10000001 01100100 01011100  
00010111 00001010 00001110  
11111010 10110001 10110001
```

Machine Readable
Machine Code

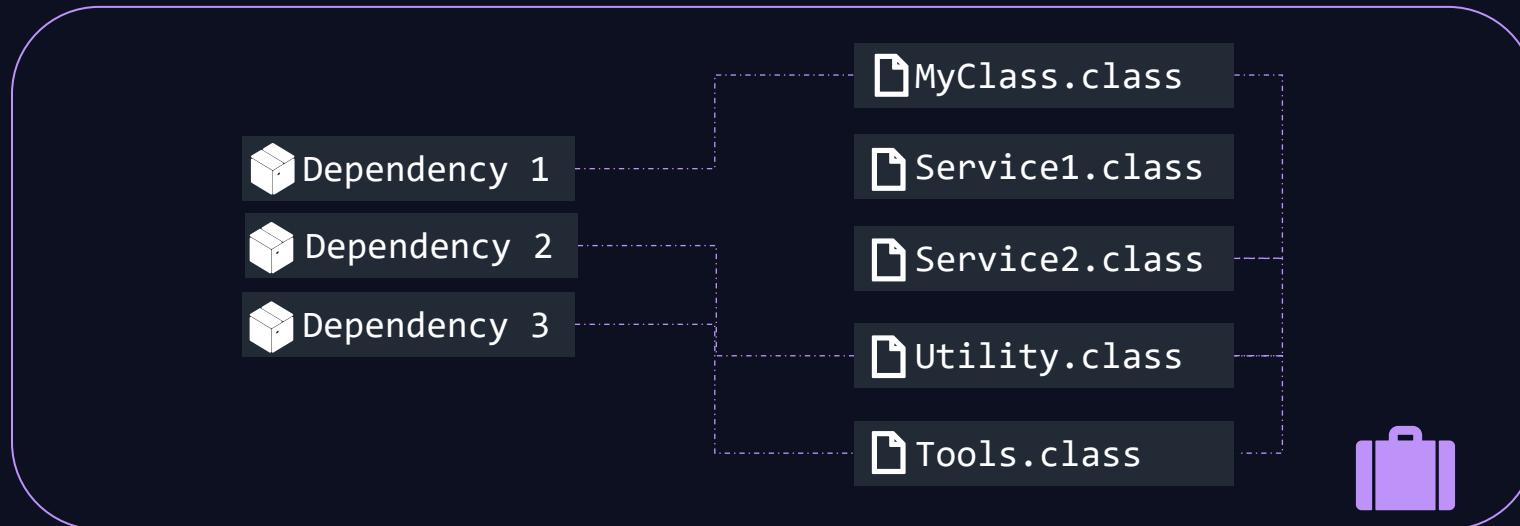
▶ javac MyClass.java

MyClass.class

▶ java MyClass

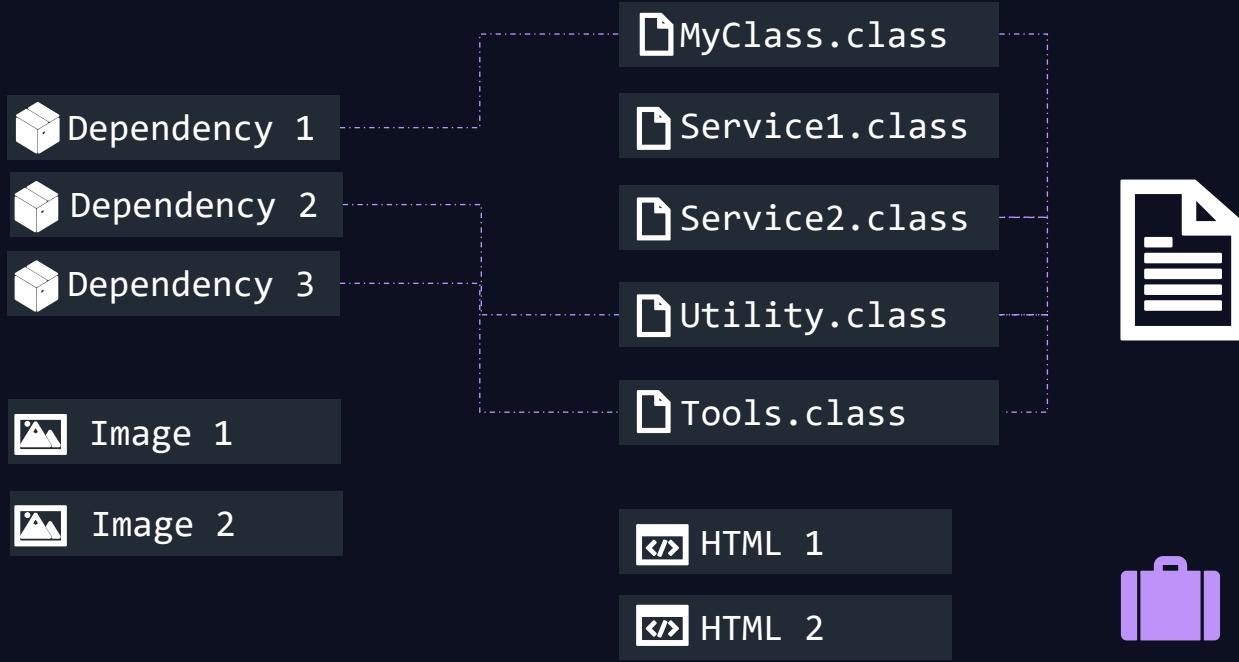
Hello World

Package



Java Archive
(JAR)

Package



META-INF/MANIFEST.MF

```
Manifest-Version: 1.0  
Created-By: 1.8.0_242 (Private Build)  
Main-Class: MyClass
```

Java Archive Web Archive
(JAR) (WAR)

```
▶ jar cf MyApp.jar MyClass.class Service1.class Service2.class ...  
MyApp.jar
```

```
▶ java -jar MyApp.jar  
Hello World
```

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

Document

```
▶ javadoc -d doc MyClass.java
```

The screenshot shows a Java API documentation page for the class `MyClass`. The page has a header with tabs for PACKAGE, CLASS (which is selected), TREE, DEPRECATED, INDEX, and HELP. Below the tabs are links for PREV CLASS, NEXT CLASS, FRAMES, and NO FRAMES. The SUMMARY section includes links for NESTED, FIELD, CONSTR, and METHOD. The DETAIL section also includes links for FIELD, CONSTR, and METHOD.

Class MyClass

java.lang.Object
MyClass

public class **MyClass**
extends java.lang.Object

Prints Hello World Message

Constructor Summary

Constructors

Constructor and Description

MyClass()

Build Process



Develop



Compile



Package



Document

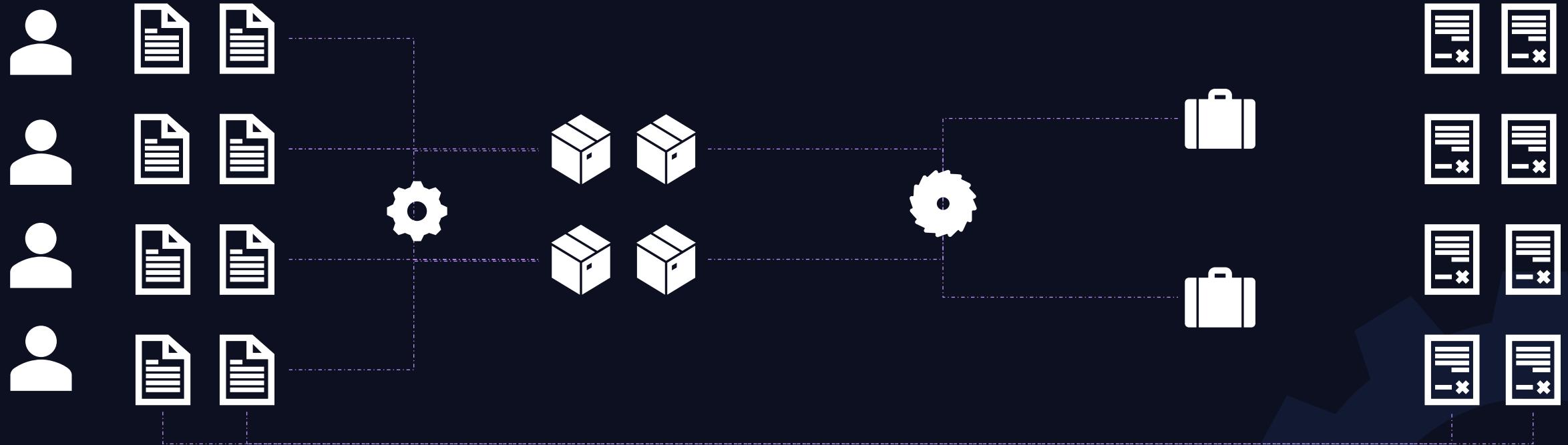
```
▶ javac MyClass.java
```

```
▶ jar cf MyClass.jar ..
```

```
▶ javadoc MyClass.java
```

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

Build Process



Build Tools

- Maven
- Gradle
- ANT

```
# Build Steps
```

```
1.Compile
```

```
2.Package
```

```
3.Document
```

ANT

► ant compile jar

BUILD SUCCESSFUL

Total time: 2 seconds

► javac MyClass.java

► javadoc MyClass.java

► jar cf MyClass.jar ..

build.xml

```
<?xml version="1.0"?>
<project name="Ant" default="main" basedir=".">
    <!-- Compiles the java code (including the usage of library for JUnit) -->
    <target name="compile">
        <javac srcdir="/app/src" destdir="/app/build">
        </javac>
    </target>
    <!-- Creates Javadoc -->
    <target name="docs" depends="compile">
        <javadoc packagenames="src" sourcepath="/app/src" destdir="/app/docs">
            <!-- Define which files / directory should get included, we include all -->
            <fileset dir="/app/src">
                <include name="**" />
            </fileset>
        </javadoc>
    </target>
    <!--Creates the deployable jar file -->
    <target name="jar" depends="compile">
        <jar basedir="/app/build" destfile="/app/dist/MyClass.jar" >
            <manifest>
                <attribute name="Main-Class" value="MyClass" />
            </manifest>
        </jar>
    </target>
    <target name="main" depends="compile, jar, docs">
        <description>Main target</description>
    </target>
</project>
```

Maven

Branch: 2.9.0 [shopizer / pom.xml](#)

Dima removed duplicate dependencies

4 contributors 

677 lines (592 sloc) | 20.7 KB

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>com.shopizer</groupId>
8   <artifactId>shopizer</artifactId>
9   <version>2.9.0</version>
10  <packaging>pom</packaging>
11
12  <name>shopizer</name>
13  <url>http://maven.apache.org</url>
14
15  <licenses>
16    <license>
17      <name>Apache License, Version 2.0</name>
18      <url>https://www.apache.org/licenses/LICENSE-2.0.txt</url>
19    </license>
20  </licenses>
21
```

Get the code:

Clone the repository:

```
$ git clone git://github.com/shopizer-e-commerce/shopizer
```

If this is your first time using Github, review <http://help.github.com>

You can also download the zip file containing the code

To build the application:

From the command line with Maven installed:

```
$ cd shopizer
$ mvnw clean install
```

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>
<https://github.com/shopizer-eCommerce/shopizer>

Gradle

Branch: master ▾ docker-java-sample / build.gradle

 arun-gupta upgrading the version to 3.0.6

2 contributors  

45 lines (35 sloc) | 961 Bytes

```
1 buildscript {  
2     repositories {  
3         jcenter()  
4     }  
5  
6     dependencies {  
7         classpath 'com.bmuschko:gradle-docker-plugin:3.0.6'  
8     }  
9 }  
10  
11 apply plugin: 'java'  
12 apply plugin: 'application'  
13 apply plugin: 'com.bmuschko.docker-java-application'  
14  
15 import com.bmuschko.gradle.docker.tasks.container.*  
16 import com.bmuschko.gradle.docker.tasks.image.*  
17
```

Gradle

Classical

1. Build app: `./gradlew build`

2. Run app: `./gradlew run`

Summary

- Java
- Java Runtime Environment
- Java Development Kit
- Compiling a Java application
- Packaging a given application to JARs
- What are Build Tools?



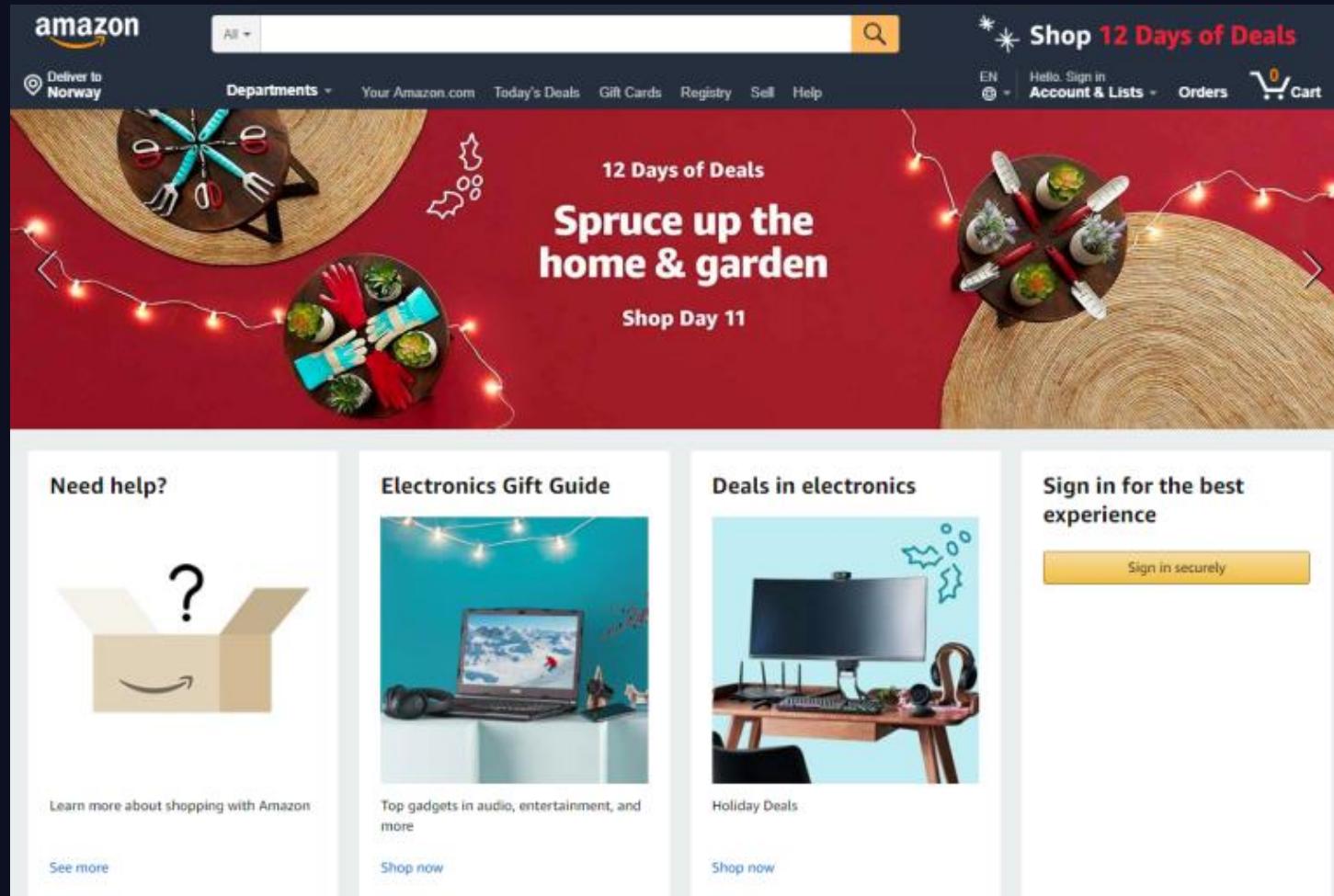
KodeKloud

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

just enough



JavaScript



jQuery



NodeJS

- Free
- Open source
- Cross Platform Compatible



[Node.js 13.x](#)

[Node.js 12.x](#)

[Node.js 11.x](#)

[Node.js 10.x](#)

[Node.js 9.x](#)

[Node.js 8.x](#)

[Node.js 7.x](#)

[Node.js 6.x](#)

[Node.js 5.x](#)

[Node.js 4.x](#)

[Node.js 0.12.x](#)

[Node.js 0.10.x](#)

Check out our full course on DevOps Pre-Requisites: <https://kodekloud.com/courses/devops-pre-requisites>

Install NodeJS

NodeSource Node.js Binary Distributions



NODESOURCE



ubuntu®



redhat.

centOS

fedora^f

```
▶ curl -sL https://rpm.nodesource.com/setup_13.x | bash -
```

```
▶ yum install nodejs
```

NodeJS Commands

```
▶ node -v
```

```
v13.10.1
```

```
▶ node add.js
```

```
Addition : 15
```

add.js

```
// Returns addition of two numbers
let add = function (a, b) {
    return a+b;
};

const a = 10, b = 5;

console.log("Addition : "+ add(a,b));
```



KodeKloud

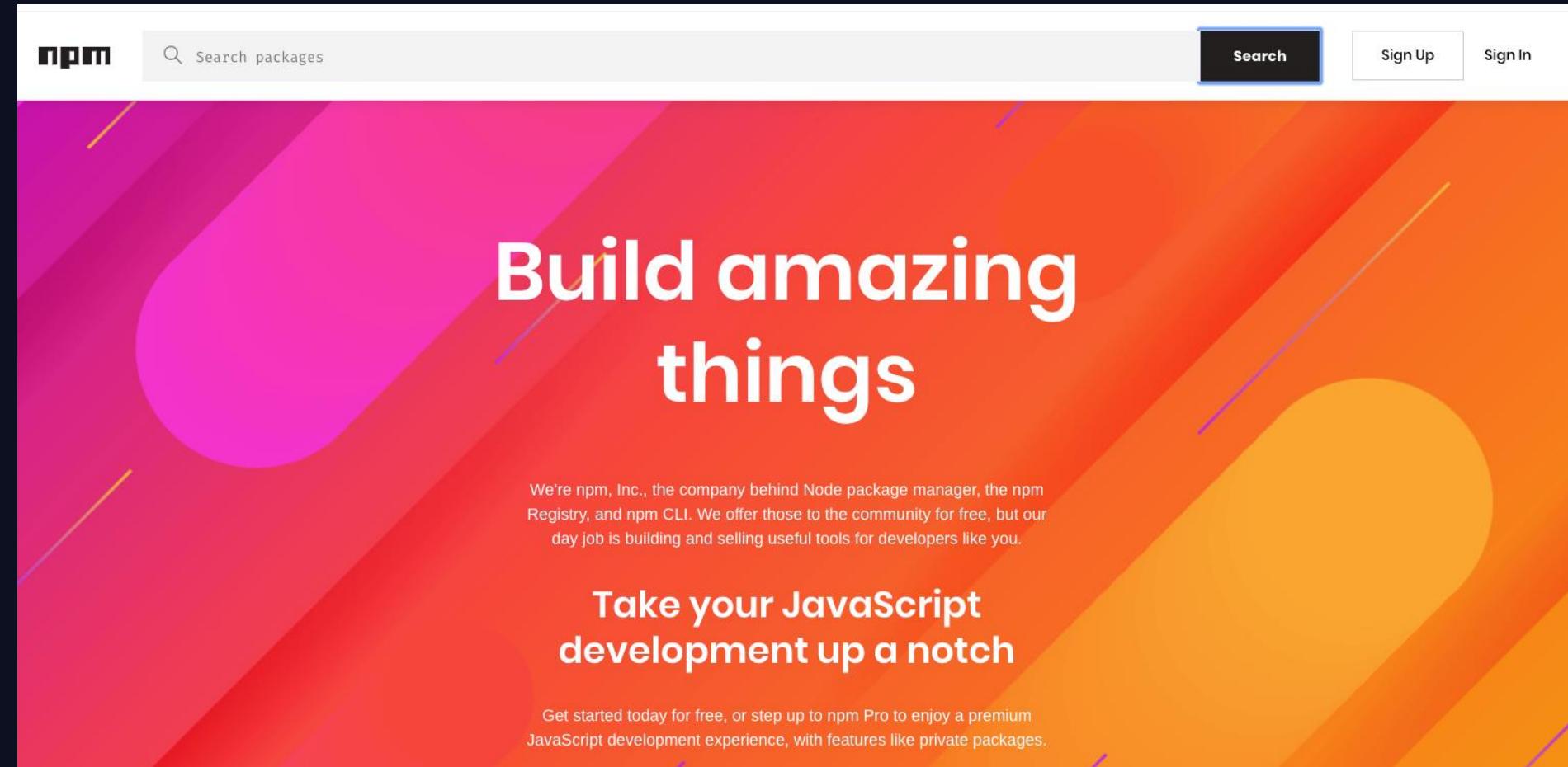
Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

just enough



Node Package Manager (NPM)

- Files
- Web Servers
- Databases
- Security
- Many More



NPM Commands

```
▶ npm -v
```

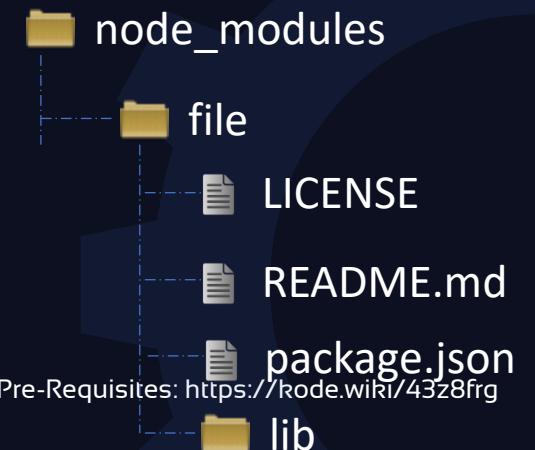
```
6.13.7
```

```
▶ npm search file
```

NAME	DESCRIPTION	AUTHOR	DATE
file	Higher level path...	=aconbere	2014-02-21
File	HTML5 FileAPI...	=coolaj86 =narf	2014-10-24
dotenv	Loads environment...	=~jcblw...	2019-10-16
fs-extra	fs-extra contains...	=jprichardson...	2019-06-28
file-loader	A file loader...	=d3viantOne...	2020-02-19

```
▶ npm install file
```

```
+ file@0.2.2  
added 1 package from 1 contributor and audited 1 package in 1.072s  
found 0 vulnerabilities
```

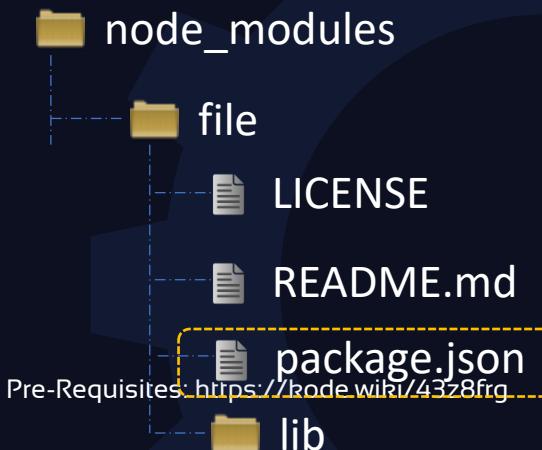


Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

NPM Commands

package.json

```
{  
  "author": {  
    "name": "Anders Conbere",  
    "email": "aconbere@gmail.com"  
  },  
  "bundleDependencies": false,  
  "devDependencies": {  
    "mocha": "1.9.x"  
  },  
  "directories": {  
    "lib": "lib"  
  },  
  "homepage": "https://github.com/aconbere/node-file-utils#readme",  
  "license": "MIT",  
  "main": "./lib/file",  
  "name": "file",  
  "repository": {  
    "type": "git",  
    "url": "git+ssh://git@github.com/aconbere/node-file-utils.git"  
  },  
  "tags": [  
    "file",  
    "path",  
    "fs",  
    "walk"  
  ],  
  "version": "0.2.2"  
}
```



NPM Commands

```
▶ npm install file
```

```
+ file@0.2.2  
added 1 package from 1 contributor and audited 1 package in 1.072s  
found 0 vulnerabilities
```

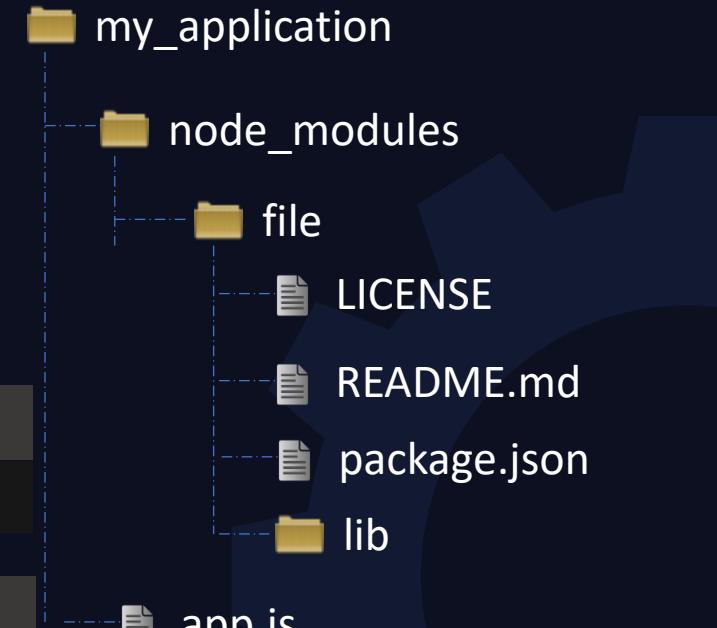
```
app.js
```

```
var file = require("file");  
  
file.mkdirs("/tmp/dir1")
```

```
▶ node -e "console.log(module.paths)"
```

```
[ '/app/node_modules', '/node_modules' ]
```

```
▶ npm install file -g
```



```
my_application
```

```
node_modules
```

```
file
```

```
LICENSE
```

```
README.md
```

```
package.json
```

```
lib
```

```
app.js
```

Common Modules

Built-In Modules		External Modules	
fs	To handle filesystem	express	Fast, unopinionated, minimalist web framework
http	To host an HTTP server	react	To create user interfaces
os	To work with the Operating System	debug	To debug applications
events	To handle events	async	To work with asynchronous JS
tls	To implement TLS and SSL	lodash	To work with arrays, objects, strings etc
url	To Parse URL Strings		

```
▶ ls /usr/lib/node_modules/npm/node_modules/
```

```
▶ ls /usr/lib/node_modules/
```

Application Dependencies

package.json

```
{  
  "name": "example-contentful-theExampleApp-js",  
  "version": "0.0.0",  
  "private": true,  
  "dependencies": {  
    "body-parser": "^1.18.2",  
    "contentful": "^6.0.0",  
    "cookie-parser": "~1.4.3",  
    "dotenv": "^5.0.0",  
    "execa": "^0.9.0",  
    "express": "^4.16.2",  
    "helmet": "^3.11.0",  
    "lodash": "^4.17.5",  
    "marked": "^0.3.16",  
    "morgan": "^1.9.1",  
    "pug": "~2.0.0-beta6"  
  }  
}
```



KodeKloud

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

just enough

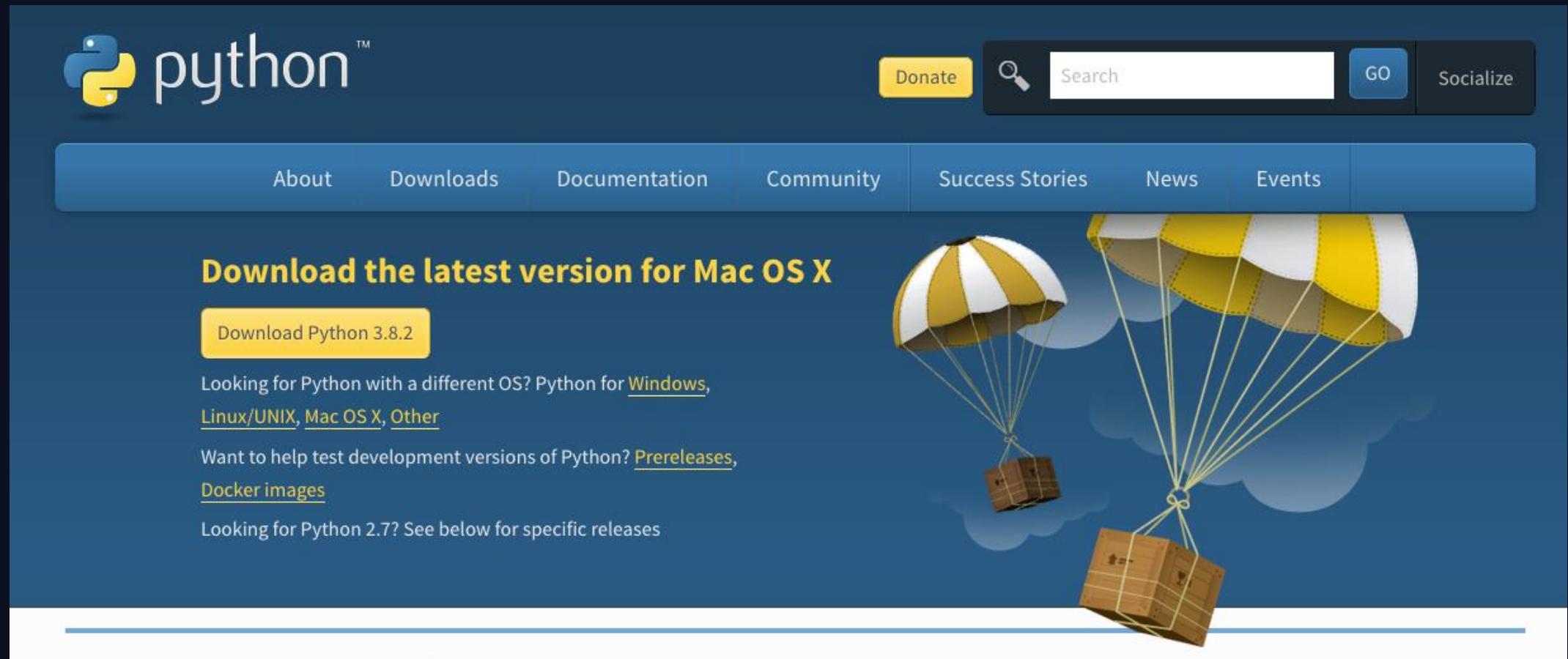


Python

- Free
- Open source
- Cross Platform Compatible



Download



The screenshot shows the Python.org website's download section. At the top, there's a navigation bar with links for About, Downloads, Documentation, Community, Success Stories, News, and Events. Below the navigation is a large yellow button labeled "Download Python 3.8.2". To the right of this button is a search bar with a magnifying glass icon and a "GO" button. There's also a "Socialize" button. A prominent graphic on the right side features two wooden crates suspended by parachutes against a blue sky with white clouds.

python™

Donate

Search

GO

Socialize

About Downloads Documentation Community Success Stories News Events

Download the latest version for Mac OS X

Download Python 3.8.2

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Prereleases](#), [Docker images](#)

Looking for Python 2.7? See below for specific releases

Versions

- Python2 – (2000 – 2010)
- Python3 – (2008 to Present)

Install

```
▶ yum install python2
```

```
▶ python2
```

```
Python 2.7.16 (default, Nov 17 2019, 00:07:27)
[GCC 8.3.1 20190507 (Red Hat 8.3.1-4)] on linux2
Type "help", "copyright", "credits" or "license"
for more information.

>>> exit()
```

```
▶ yum install python36
```

```
▶ python3
```

```
Python 3.6.8 (default, Nov 21 2019, 19:31:34)
[GCC 8.3.1 20190507 (Red Hat 8.3.1-4)] on linux
Type "help", "copyright", "credits" or "license"
for more information.

>>> exit()
```

```
▶ python2 -V
```

```
Python 2.7.16
```

```
▶ python3 -V
```

```
Python 3.6.8
```

Python Commands

```
▶ python2 main.py
```

```
Hello World
```

```
main.py
```

```
def print_message():
    print("Hello World")

if __name__ == '__main__':
    print_message()
```



KodeKloud

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

just enough



Python Package Manager (pip)

```
▶ python2 -V
```

```
Python 2.7.16
```

```
▶ pip2 -V
```

```
pip 9.0.3 from /usr/lib/python2.7/site-packages  
(python 2.7)
```

```
▶ python3 -V
```

```
Python 3.6.8
```

```
▶ pip3 -V
```

```
pip 9.0.3 from /usr/lib/python3.6/site-packages  
(python 3.6)
```

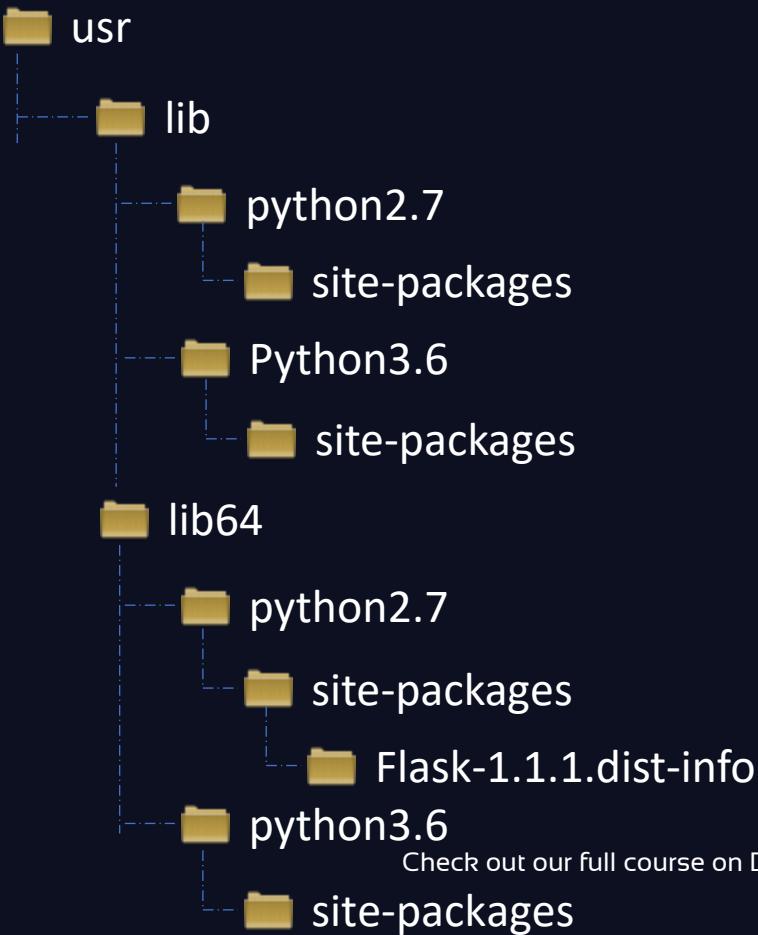
```
▶ pip -V
```

```
pip 9.0.3 from /usr/lib/python2.7/site-packages (python 2.7)
```

```
▶ pip install flask
```

Python Package Manager (pip)

```
▶ pip install flask
```



Python Package Manager (pip)

```
▶ pip install flask
```

```
▶ pip show flask
```

```
Name: Flask
Version: 1.1.1
Summary: A simple framework for building complex
web applications.
Home-page: https://palletsprojects.com/p/flask/
Author: Armin Ronacher
Author-email: armin.ronacher@active-4.com
License: BSD-3-Clause
Location: /usr/lib64/python2.7/site-packages
Requires: Werkzeug, click, Jinja2, itsdangerous
```

```
▶ python2 -c "import sys; print(sys.path)"
```

```
'/usr/lib/python27.zip', '/usr/lib64/python2.7',
'/usr/lib64/python2.7/plat-linux2',
'/usr/lib64/python2.7/lib-tk',
'/usr/lib64/python2.7/lib-old',
'/usr/lib64/python2.7/lib-dynload',
'/usr/lib64/python2.7/site-packages',
'/usr/lib/python2.7/site-packages'
```

```
main.py
from flask import Flask, request
app = Flask(__name__)
@app.route('/')
def hello():
    return 'Hello, World'
```

```
python2.7
└── site-packages
    └── Flask-1.1.1.dist-info
python3.6
└── site-packages
```

Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>

Requirements

```
► pip install flask
```

```
► pip install jinja2
```

```
► pip install markupsafe
```

```
► pip install Werkzeug
```

```
► pip install requests
```

```
► pip install gunicorn
```

```
► pip install flask jinja2 markupsafe
```

```
requirements.txt
```

```
Flask==0.10.1
```

```
Jinja2==2.7.3
```

```
MarkupSafe==0.23
```

```
Werkzeug==0.9.6
```

```
requests==2.3.0
```

```
gunicorn==18.0
```

```
► pip install -r requirements.txt
```

Upgrade/Uninstall Package

```
▶ pip install flask --upgrade
Installing collected packages: click, flask
  Attempting uninstall: flask
    Found existing installation: Flask 0.10.1
    Uninstalling Flask-0.10.1:
      Successfully uninstalled Flask-0.10.1
Successfully installed click-7.1.1 flask-1.1.1
```

```
▶ pip uninstall flask
Found existing installation: Flask 1.1.1
Uninstalling Flask-1.1.1:
  Would remove:
    /home/vagrant/.local/bin/flask
    /home/vagrant/.local/lib/python3.5/site-
packages/Flask-1.1.1.dist-info/*
    /home/vagrant/.local/lib/python3.5/site-
packages/flask/*
Proceed (y/n)? y
Successfully uninstalled Flask-1.1.1
```

Other Package Managers

- easy_install

► easy_install install app



app.py



✗ setuptools



app.egg

- wheels

► pip install app.whl



app.py



✗ setuptools



app.whl

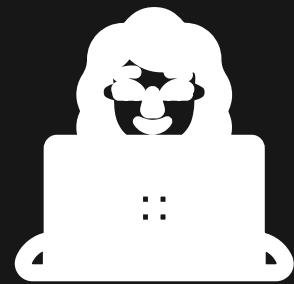
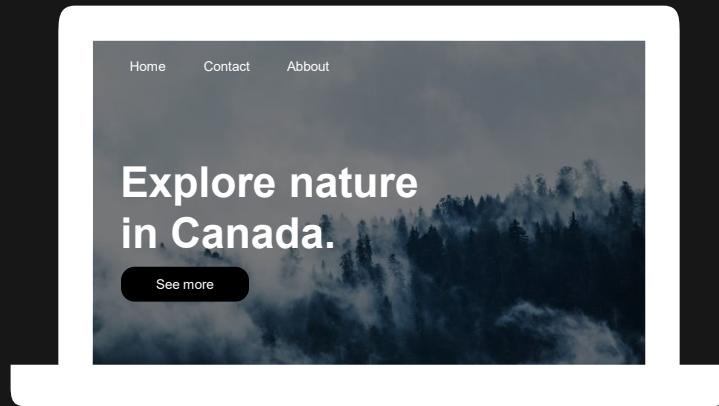


KodeKloud

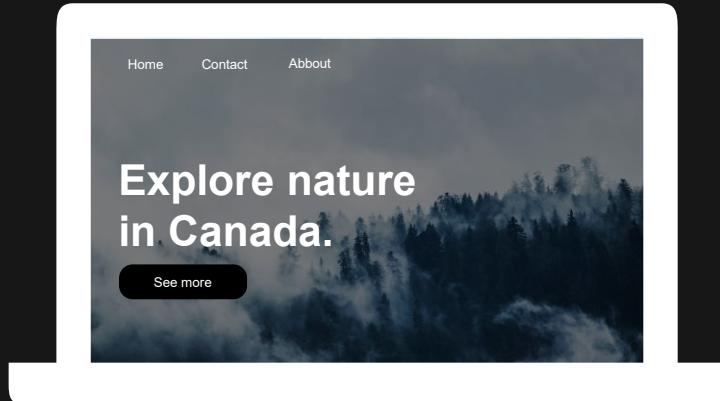
Check out our full course on DevOps Pre-Requisites: <https://kode.wiki/43z8frg>



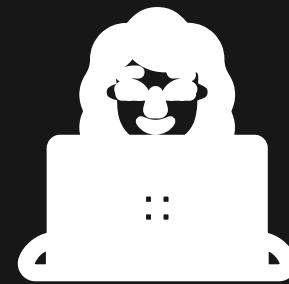
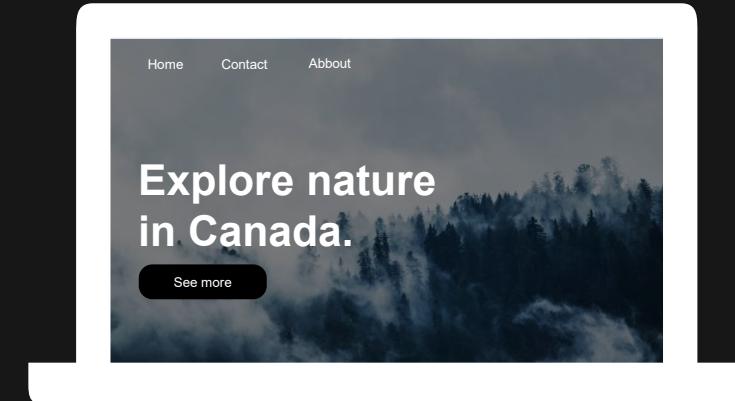
Git



www.mywebsite.com



local



www.mywebsite.com

Home Contact About

Explore nature
in Canada.

See more

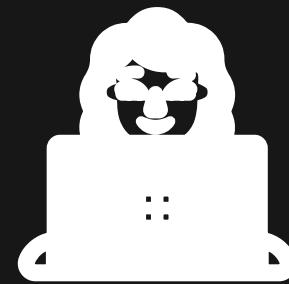
local

Home Contact About

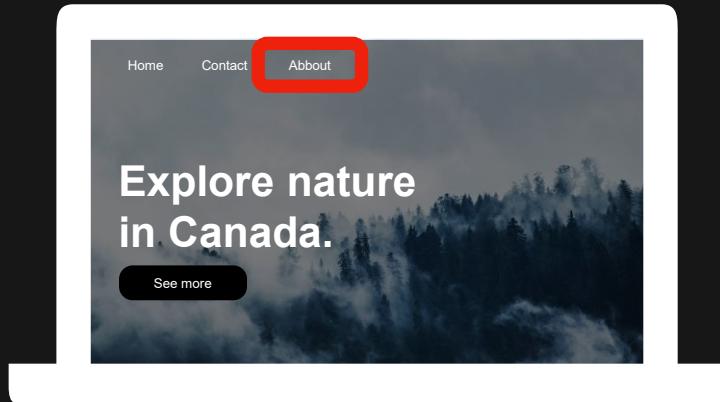
Explore nature
in Canada.

Visit Canada's lakes and
mountains.

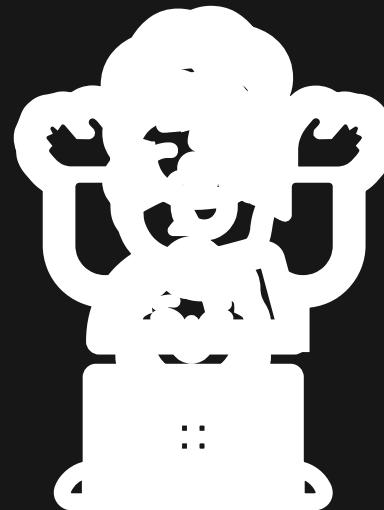
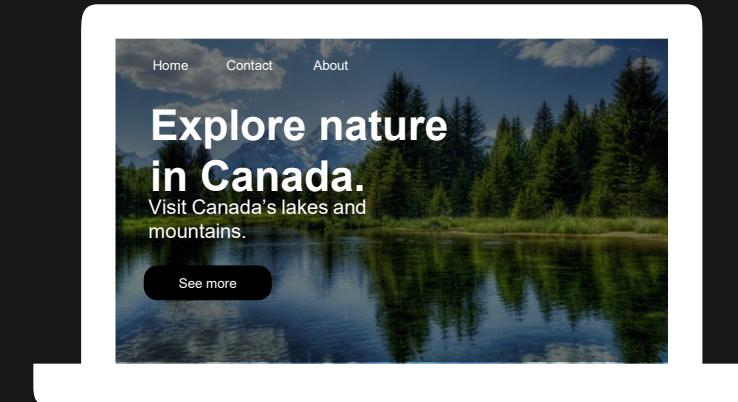
See more



www.mywebsite.com

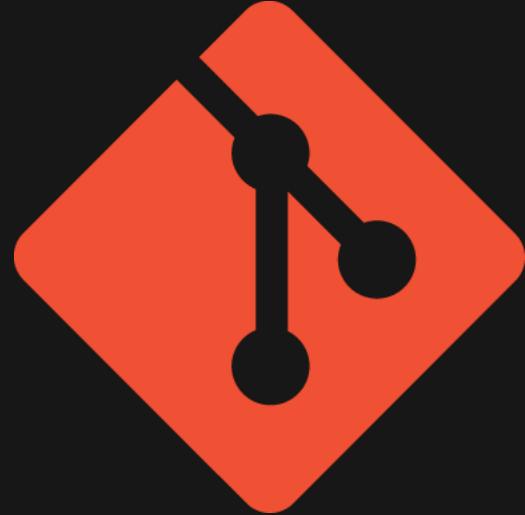


local





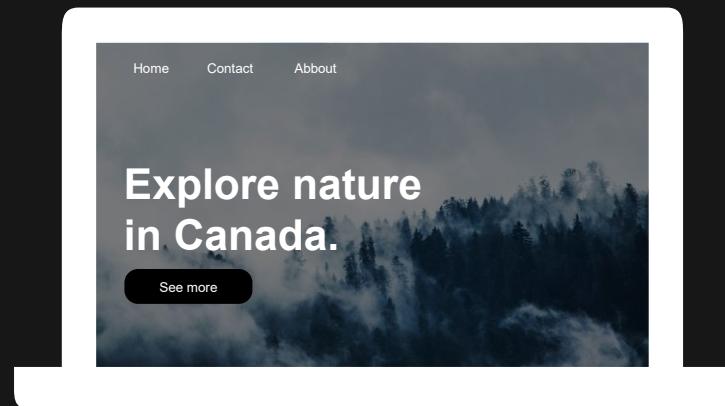
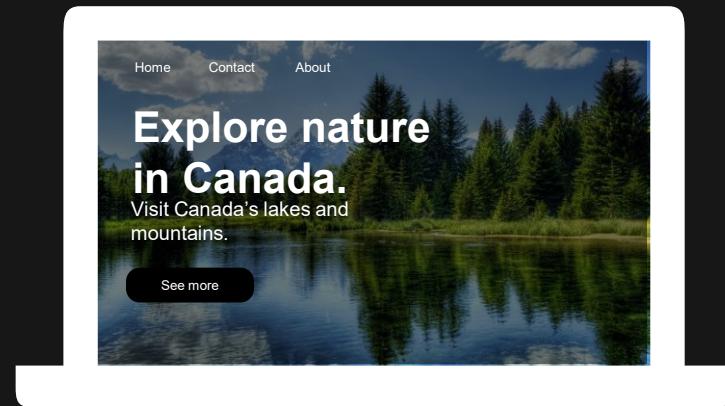
Git



Git

Content tracker

Distributed Version Control System

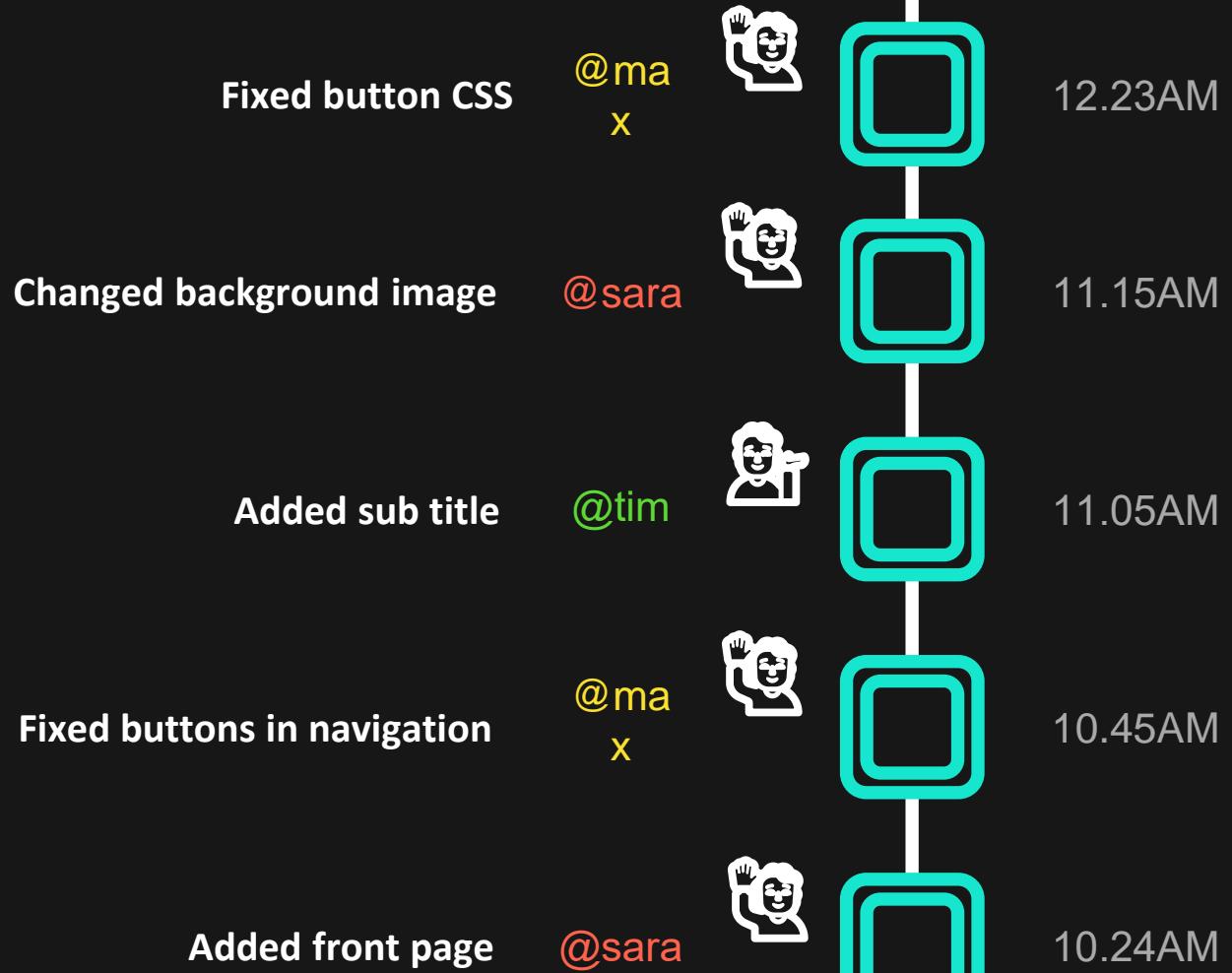


New state of your website with
the local changes



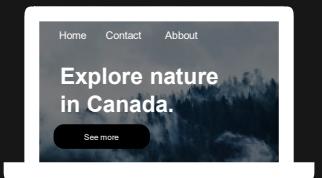
Hosted state of your website



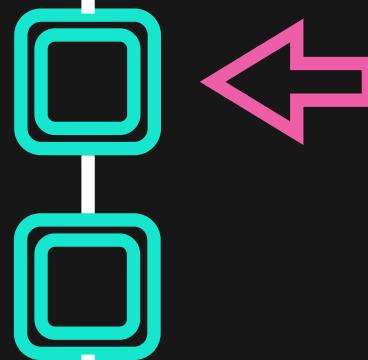




Change background

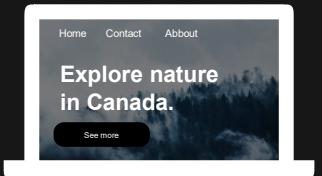


Added front page





Change background

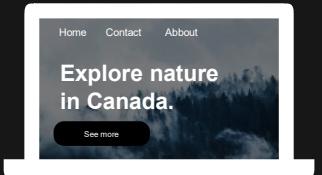


Added front page





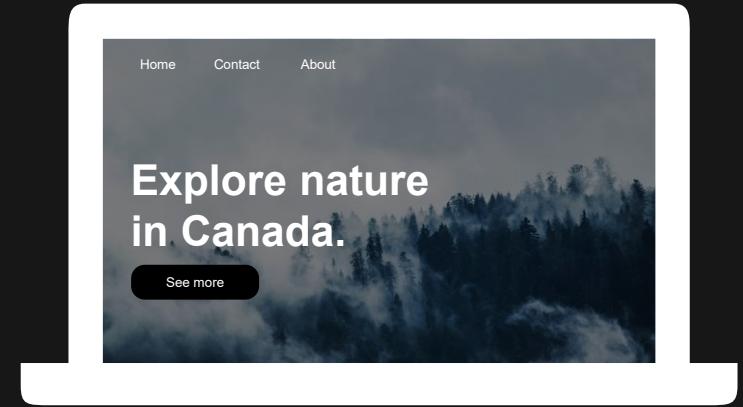
Change background



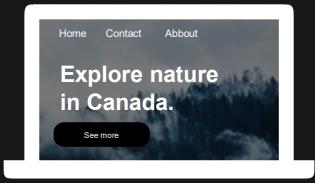
Added front page



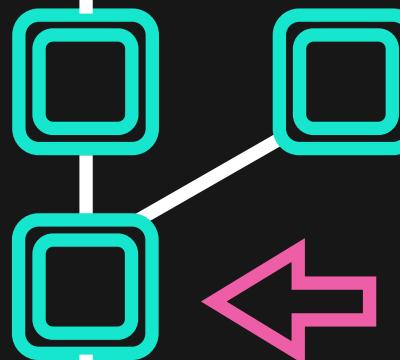
www.mywebsite.com



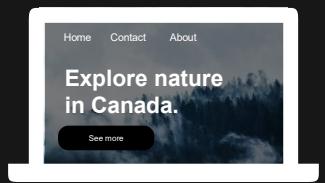
Change background



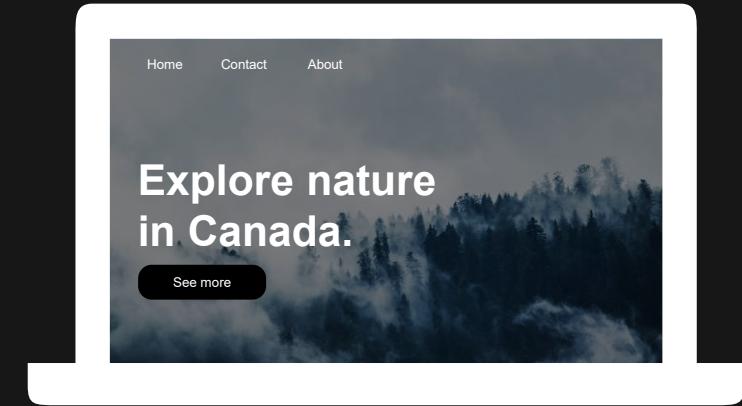
Added front page



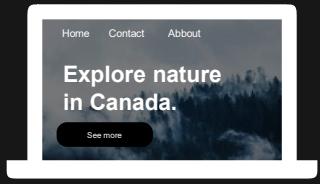
Fixed button typo



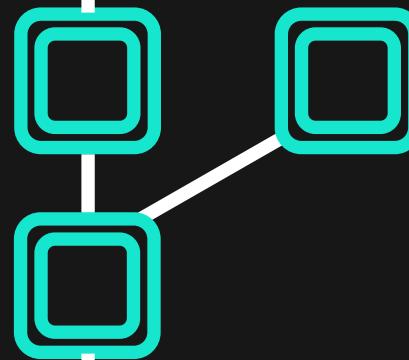
www.mywebsite.com



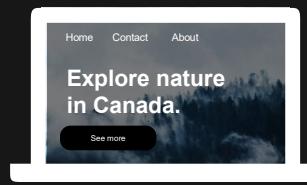
Change background



Added front page



Fixed button typo



Remote Repository



Local Repository



```
<html>
<body>
<h1>My Website!</h1>
</body>
</html>
```

Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>

Remote Repository

```
<html>
<body>
<h1>My Website!</h1>
</body>
</html>
```



Local Repository

```
<html>
<body>
<h1>My Website!</h1>
</body>
</html>
```



Local Repository



Remote Repository

```
<html>
<body>
<h1>My Website!</h1>
</body>
</html>
```



Local Repository

```
<html>
<body>
<h1>My Website!</h1>
</body>
</html>
```

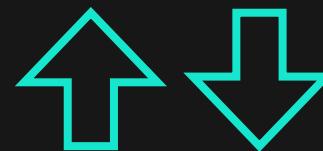


Local Repository



Remote Repository

```
<html>
<body>
<h1>My Website!</h1>
</body>
</html>
```



Local Repository

```
<html>
<body>
<h1>My Website!</h1>
</body>
</html>
```



Local Repository

```
<html>
<body>
<h1>My Website!</h1>
<p>This is some more text!</p>
</body>
</html>
```



Remote Repository

```
<html>
<body>
<h1>My Website!</h1>
</body>
</html>
```



Local Repository

```
<html>
<body>
<h1>My Website!</h1>
</body>
</html>
```



Local Repository

```
<html>
<body>
<h1>My Website!</h1>
<p>This is some more text!</p>
</body>
</html>
```



Remote Repository

```
<html>
<body>
<h1>My Website!</h1>
<p>This is some more text!</p>
</body>
</html>
```



Local Repository

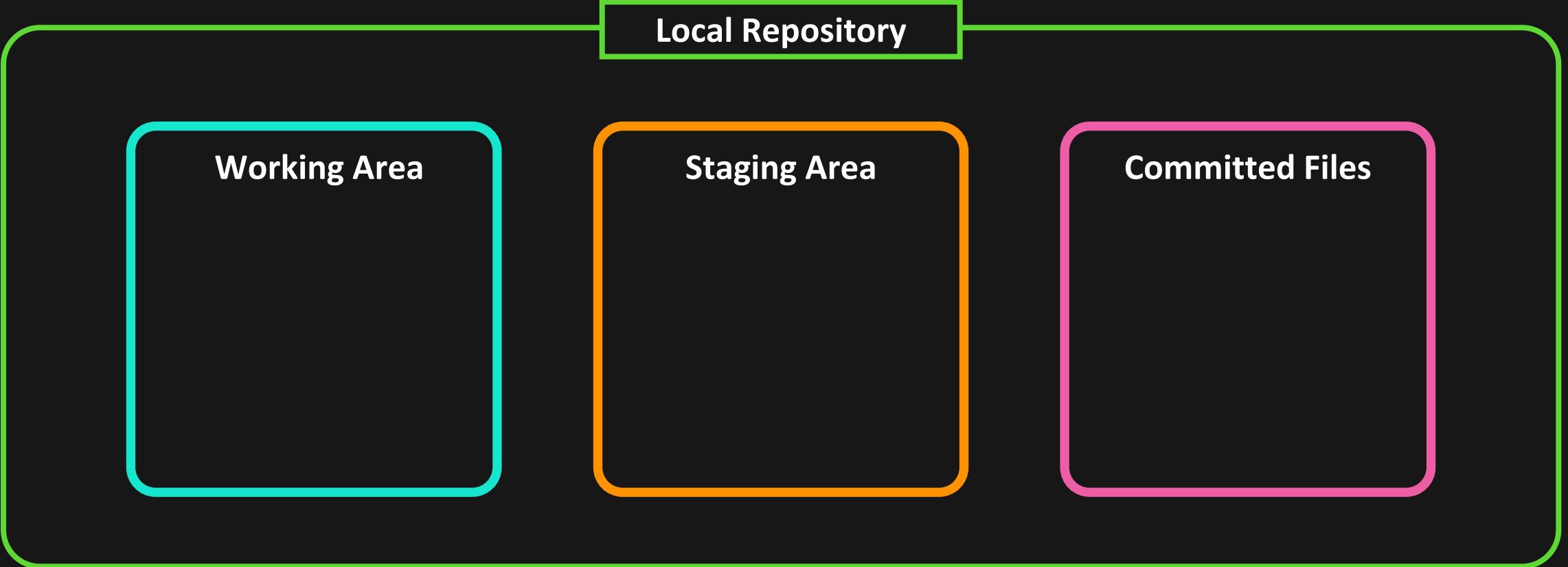
```
<html>
<body>
<h1>My Website!</h1>
</body>
</html>
```

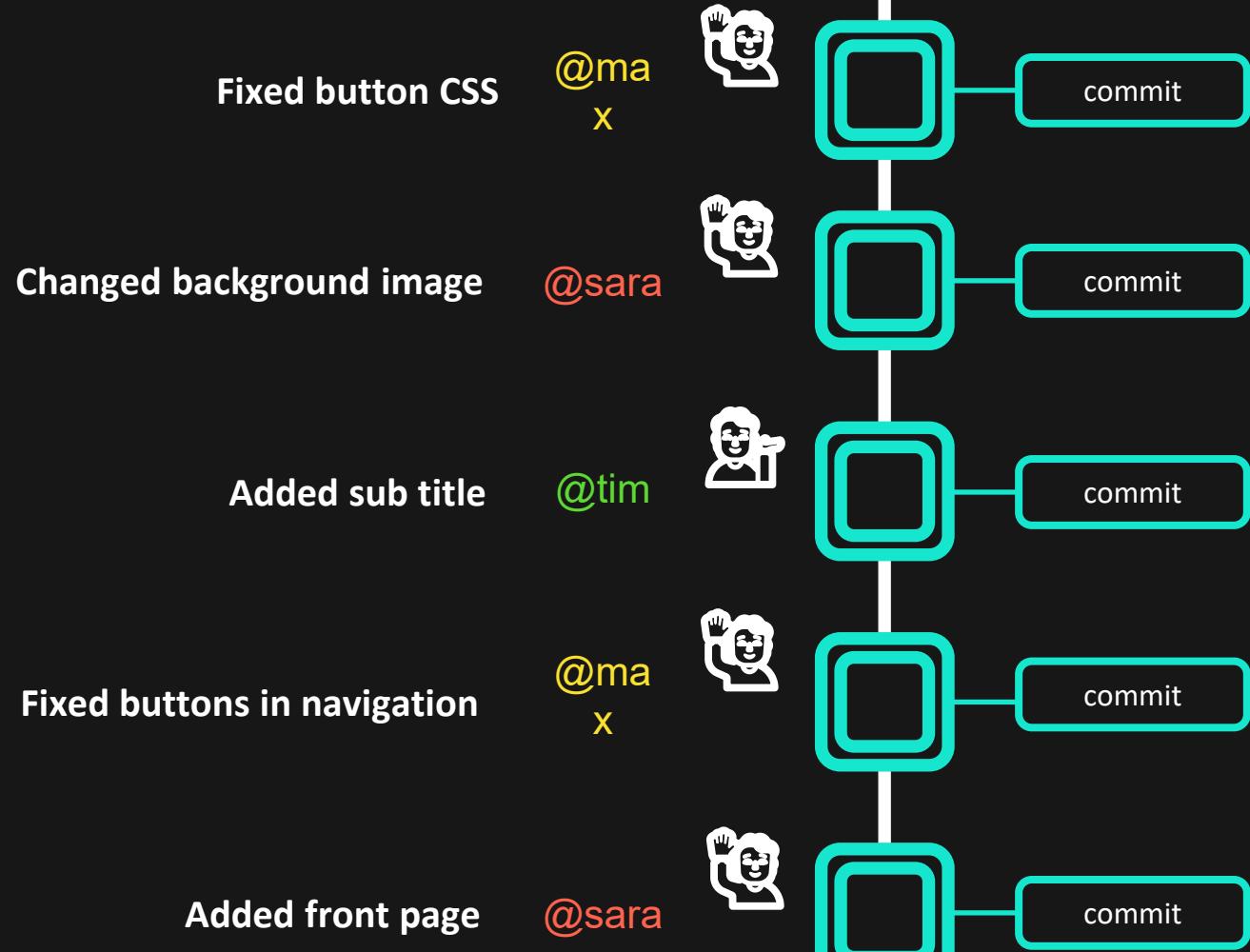


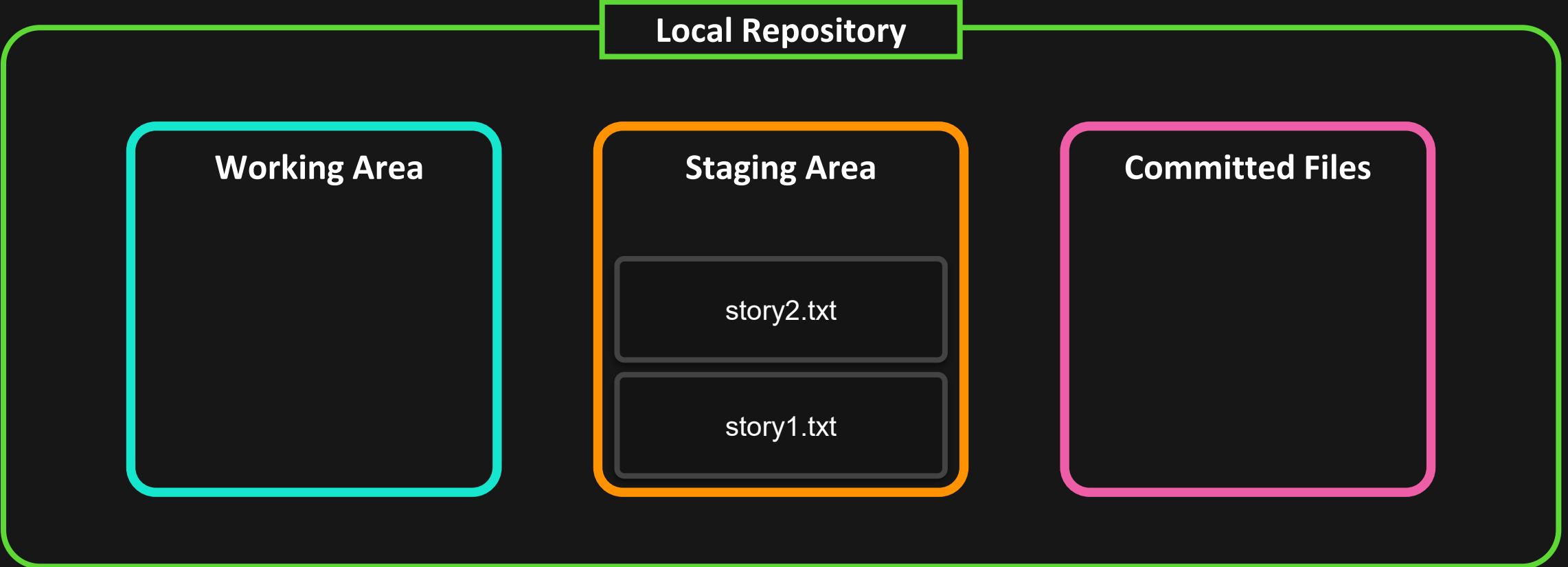
Local Repository

```
<html>
<body>
<h1>My Website!</h1>
<p>This is some more text!</p>
</body>
</html>
```









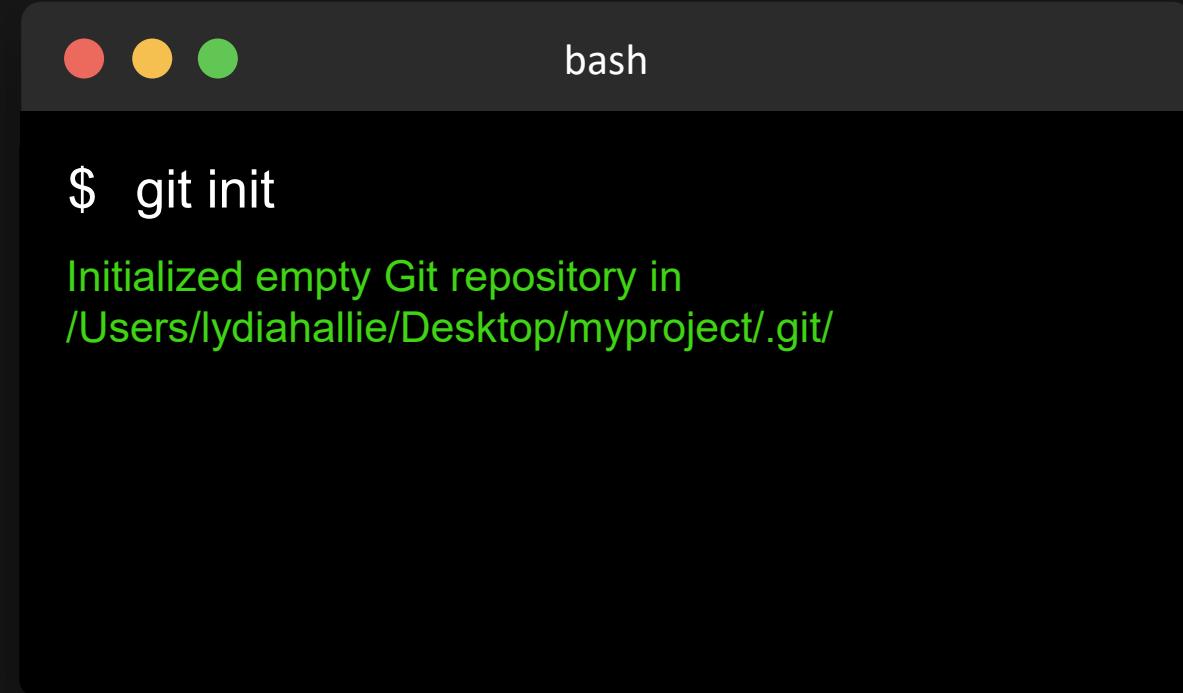


KodeKloud

Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>

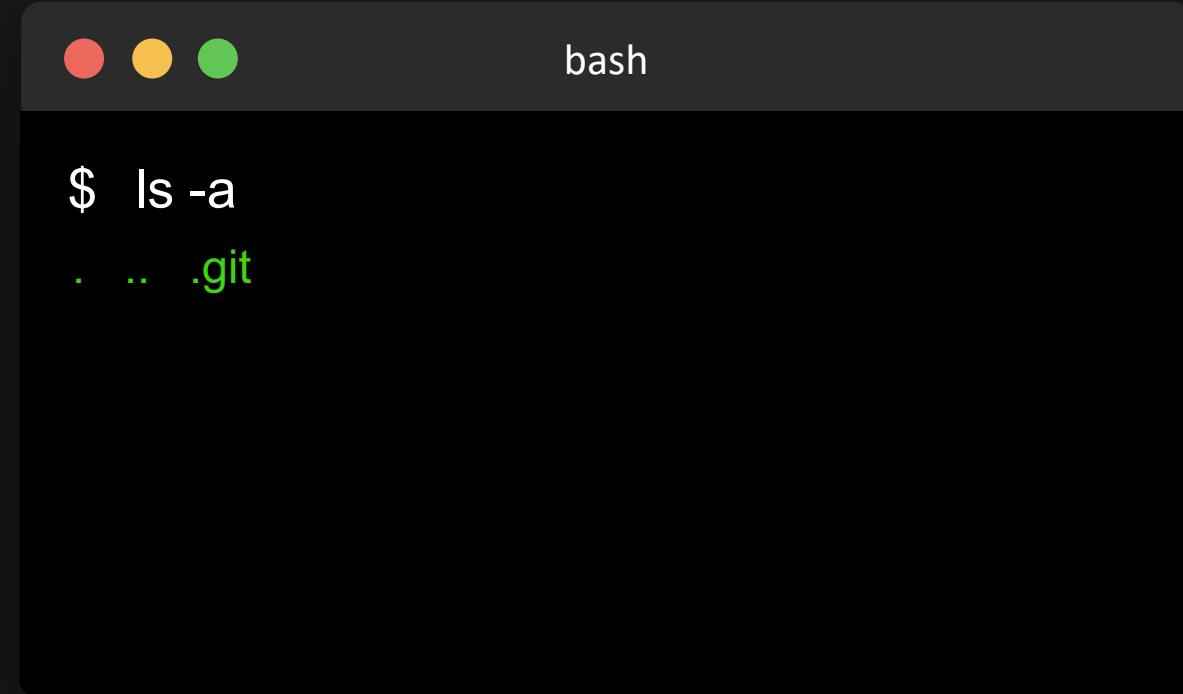


Initialize Git Repo



The image shows a dark-themed terminal window titled "bash". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar contains the word "bash". Inside the terminal, the command "\$ git init" is typed in white text. Below it, the output of the command is displayed in green text: "Initialized empty Git repository in /Users/lydiahallie/Desktop/myproject/.git/".

```
$ git init
Initialized empty Git repository in /Users/lydiahallie/Desktop/myproject/.git/
```



A screenshot of a terminal window titled "bash". The window has a dark background with light-colored text. The title bar features three colored circles (red, yellow, green) on the left and the word "bash" on the right. The main area of the terminal shows the command "\$ ls -a" followed by its output: ". . .git". The ".git" folder name is highlighted in green.

```
$ ls -a
. .. .git
```



bash

```
$ touch story1.txt  
$ echo "This is a beautiful story" >> story1.txt
```



bash

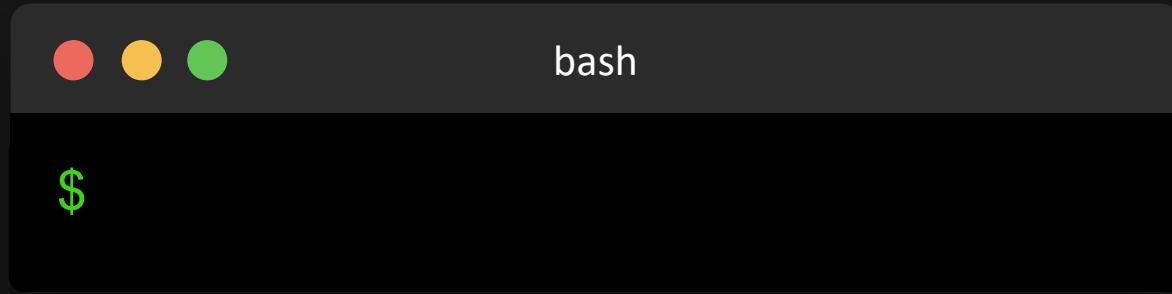
```
$ git status
```

On branch master

No commits yet

Untracked files:
story1.txt

nothing added to commit but untracked files present



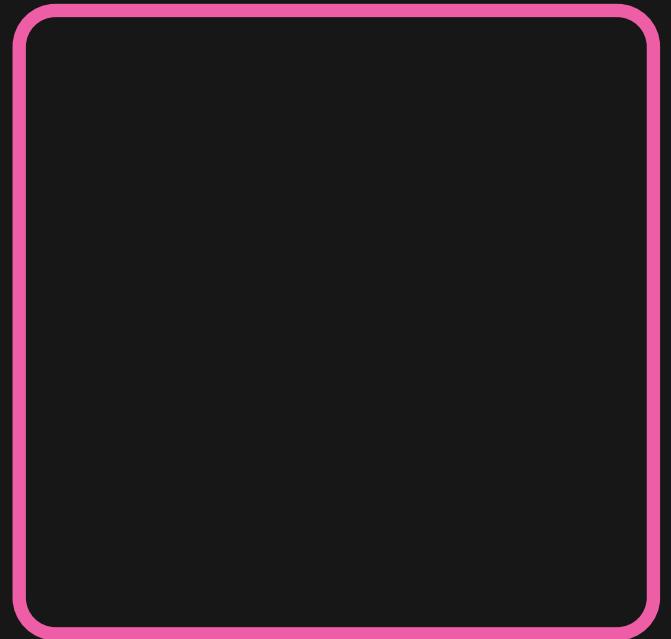
Working Area



Staging Area



Committed Files





```
bash
$ git add story1.txt
```

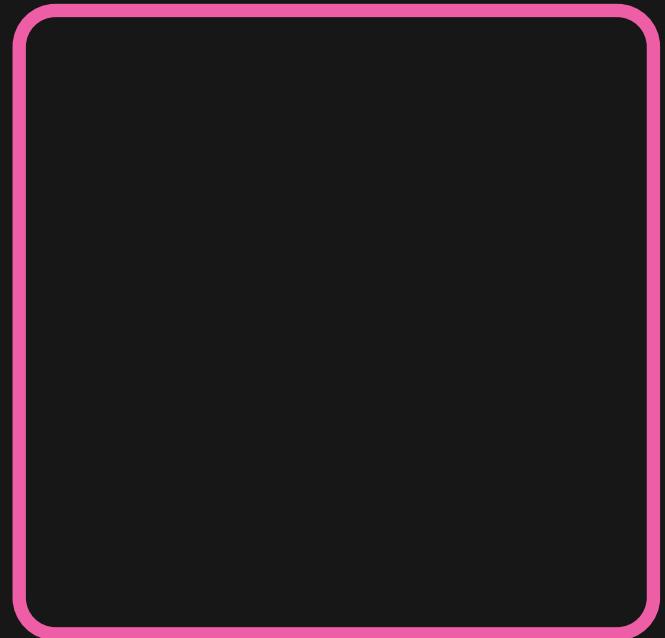
Working Area



Staging Area



Committed Files



```
$ git commit -m "Added first story"
```

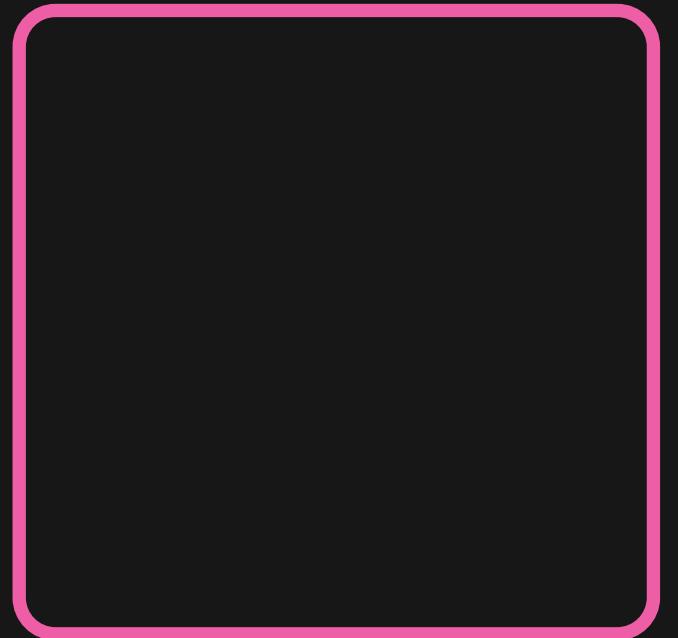
Working Area



Staging Area



Committed Files



Added first story @
me



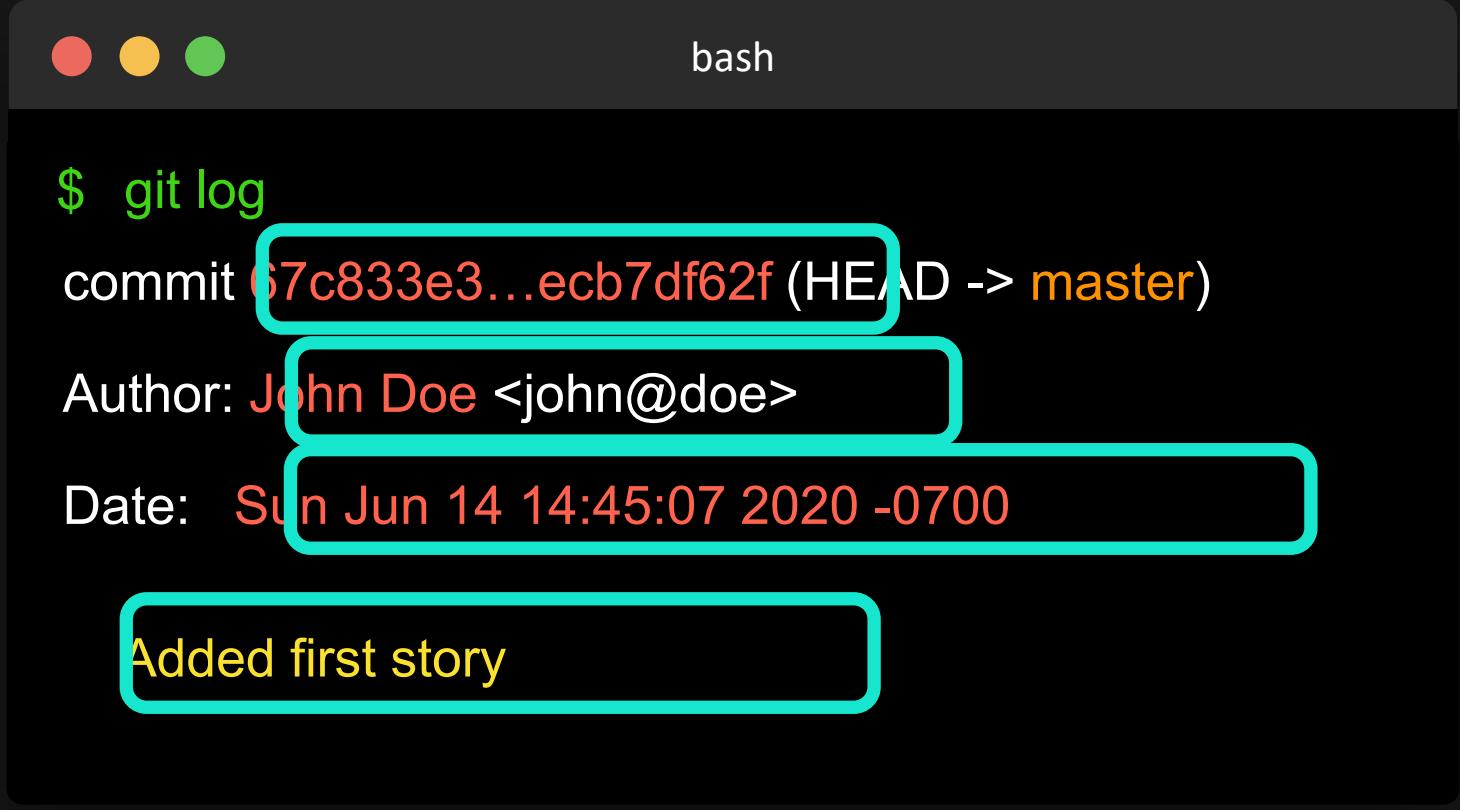


KodeKloud

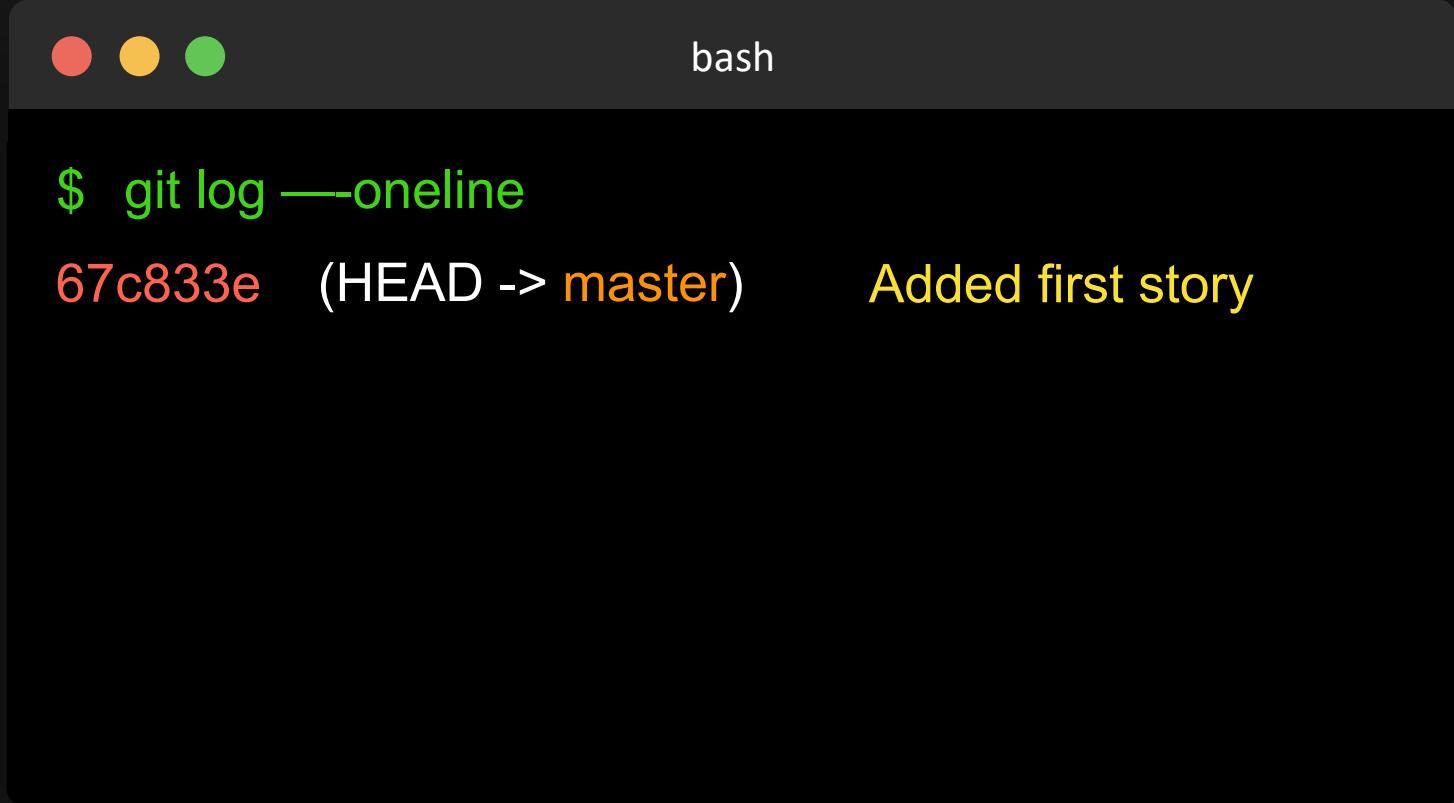
Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>



Git log



```
bash
$ git log
commit 67c833e3...ecb7df62f (HEAD -> master)
Author: John Doe <john@doe>
Date:  Sun Jun 14 14:45:07 2020 -0700
Added first story
```



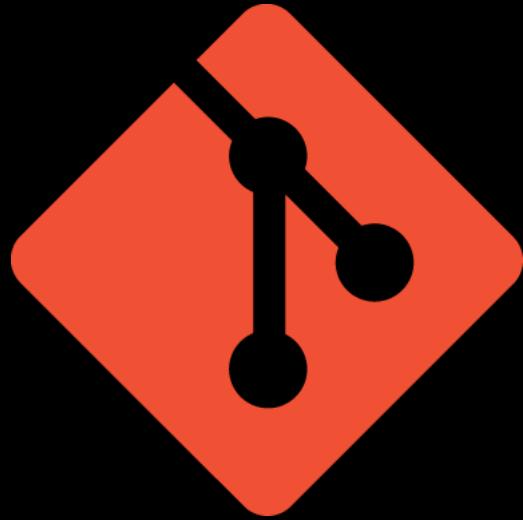
A screenshot of a dark-themed terminal window titled "bash". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar is dark with white text. Inside the terminal, the command "\$ git log --oneline" is run, followed by its output:

```
$ git log --oneline  
67c833e (HEAD -> master) Added first story
```



KodeKloud

Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>



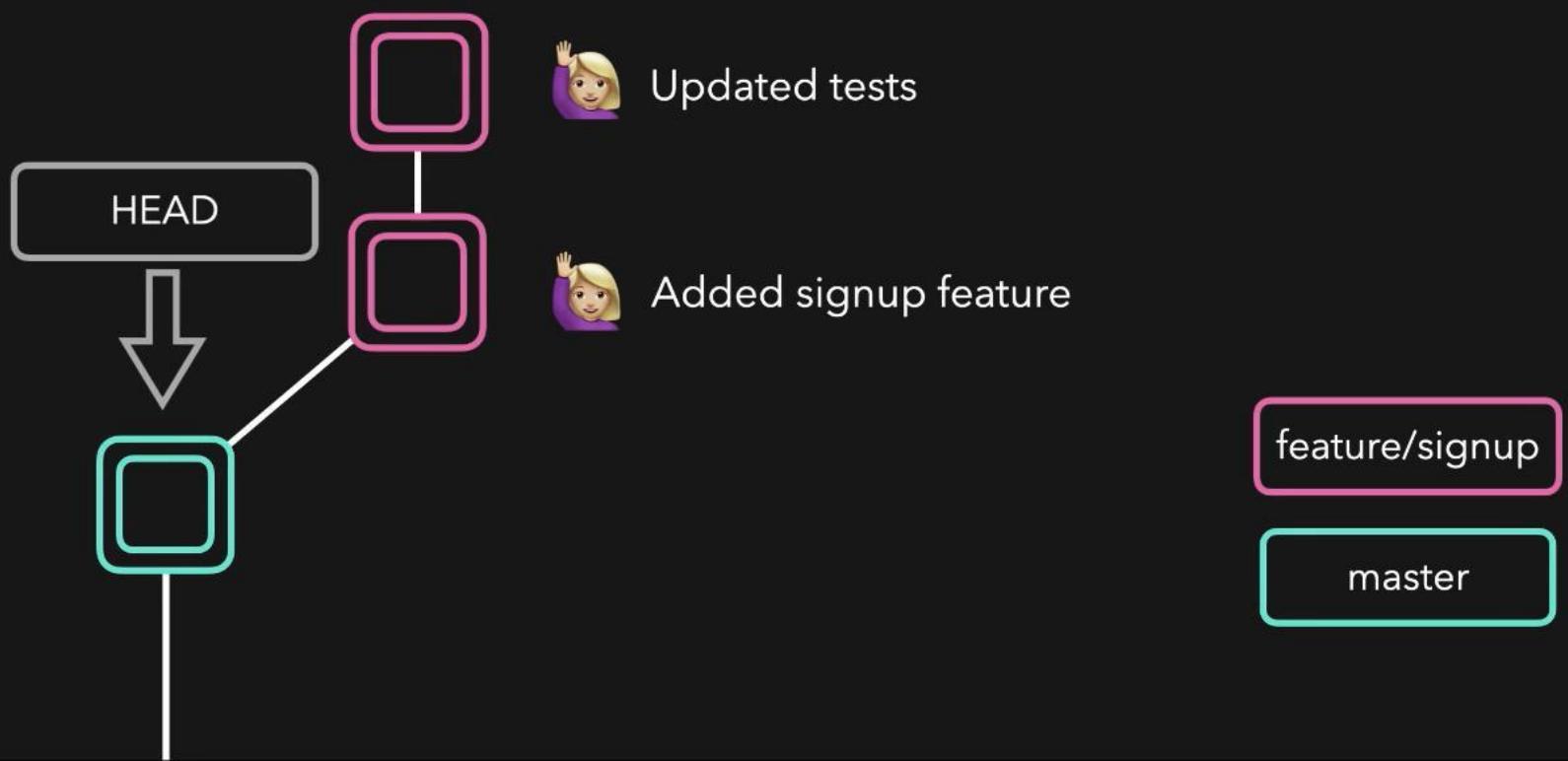
Branches

```
# Create a new branch  
$ git branch sarah  
  
# Switch to an existing branch  
$ git checkout sarah  
  
# Create a new branch and Switch to it  
$ git checkout -b max  
  
# Delete a branch  
$ git branch -d max  
  
# List all branches  
$ git branch
```



bash

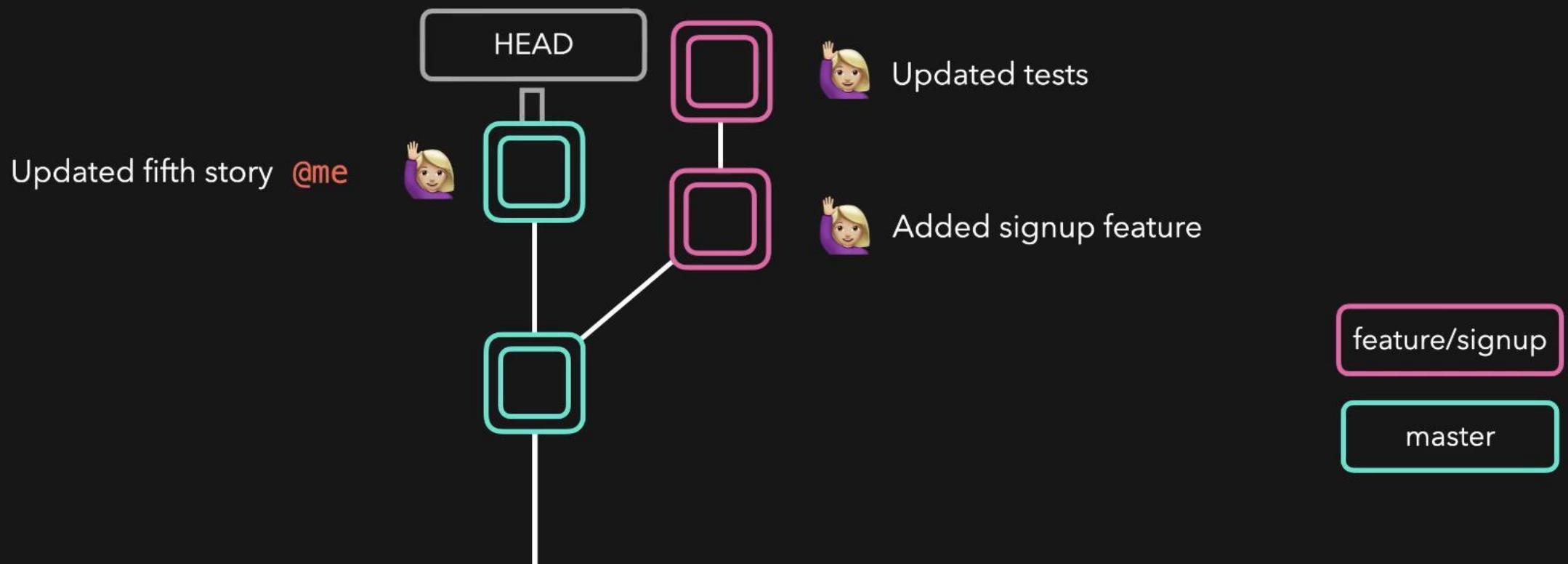
```
$ git checkout master
```





bash

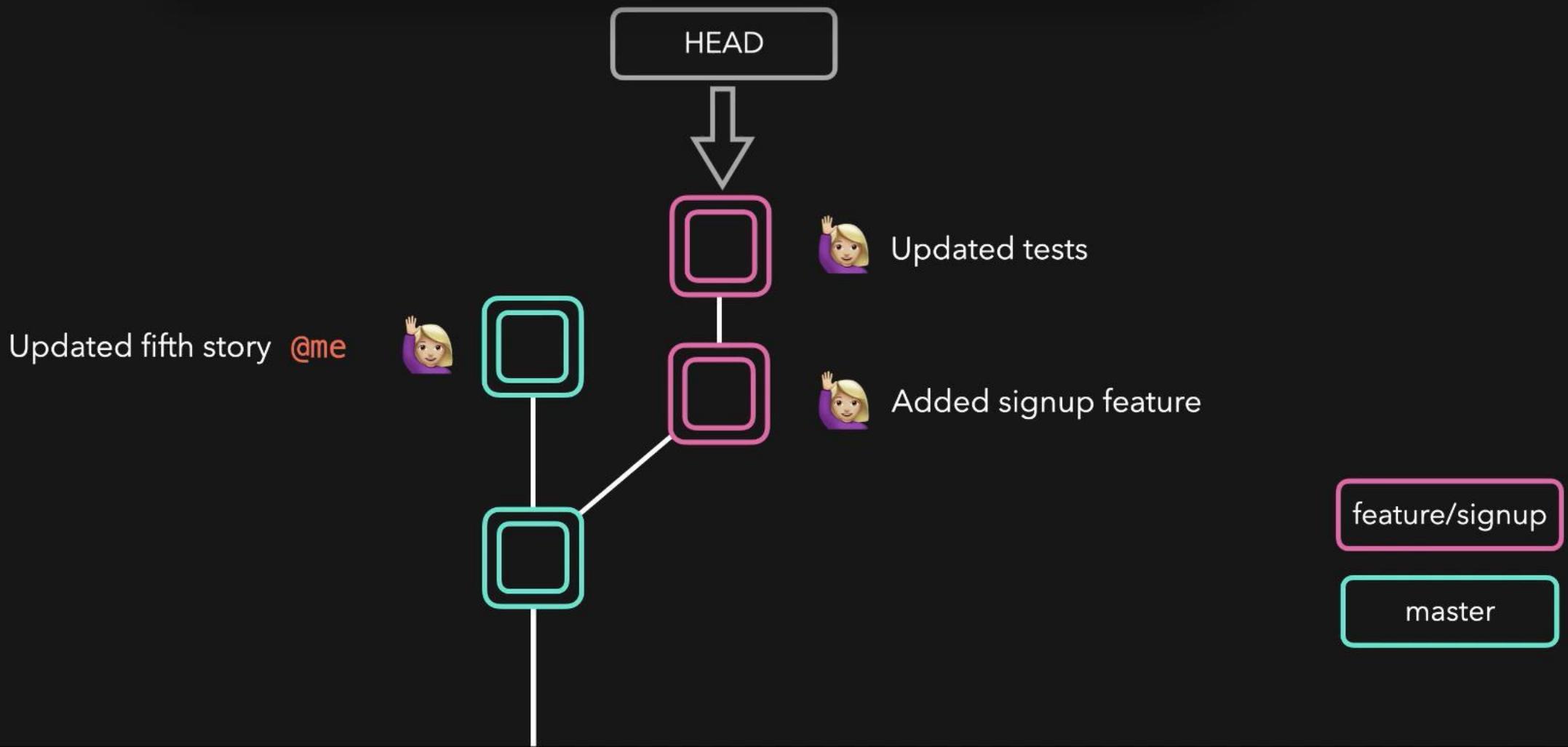
```
$ git commit -am "Updated fifth story"
```





bash

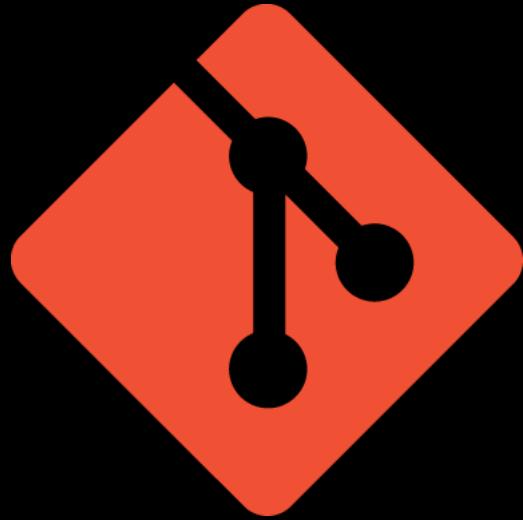
```
$ git checkout feature/signup
```



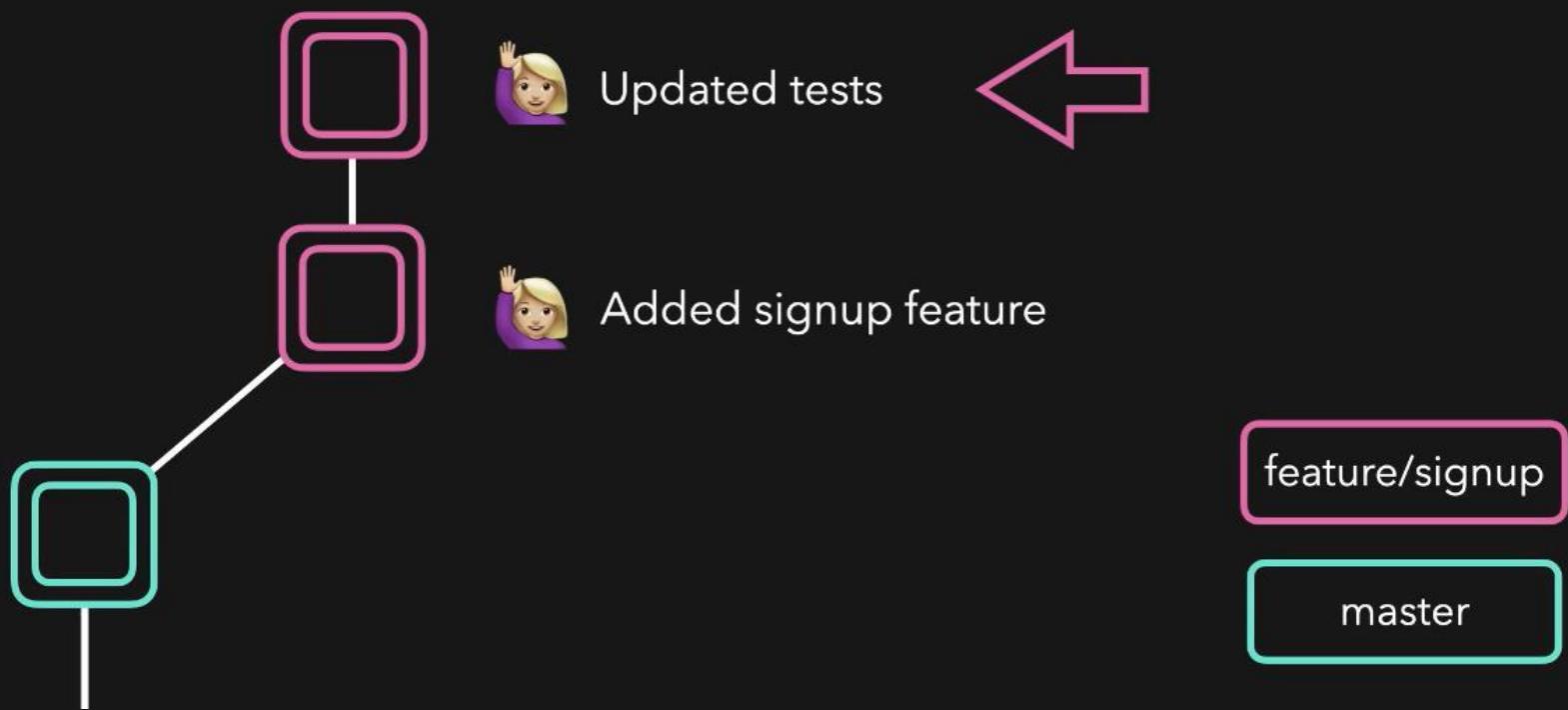


KodeKloud

Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>



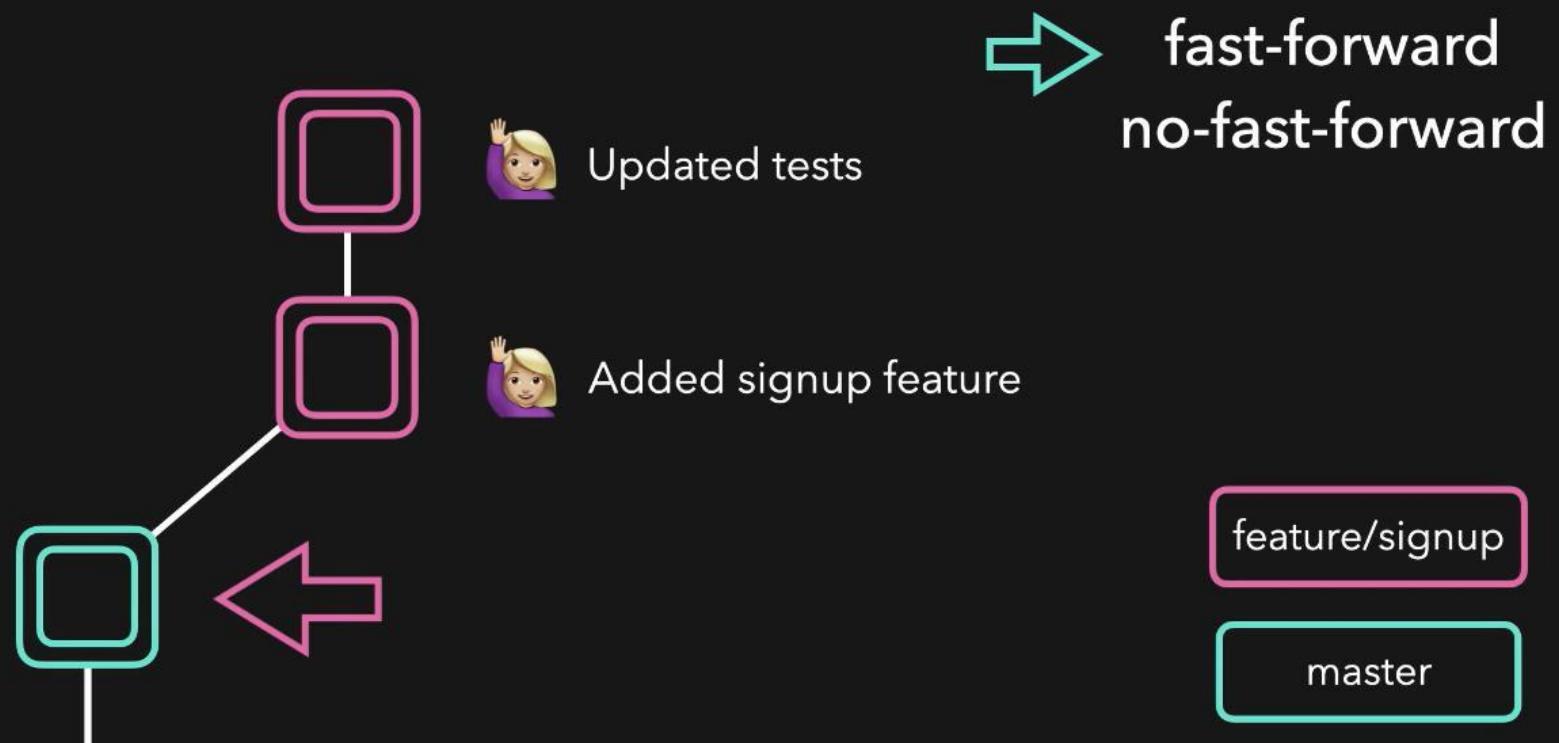
Merging





bash

```
(feature/signup)$ git checkout master  
(master)$ git merge feature/signup
```





bash

```
(feature/signup)$ git checkout master  
(master)$ git merge feature/signup
```

Merged branch `feature/signup` @me



Updated tests



Added signup feature

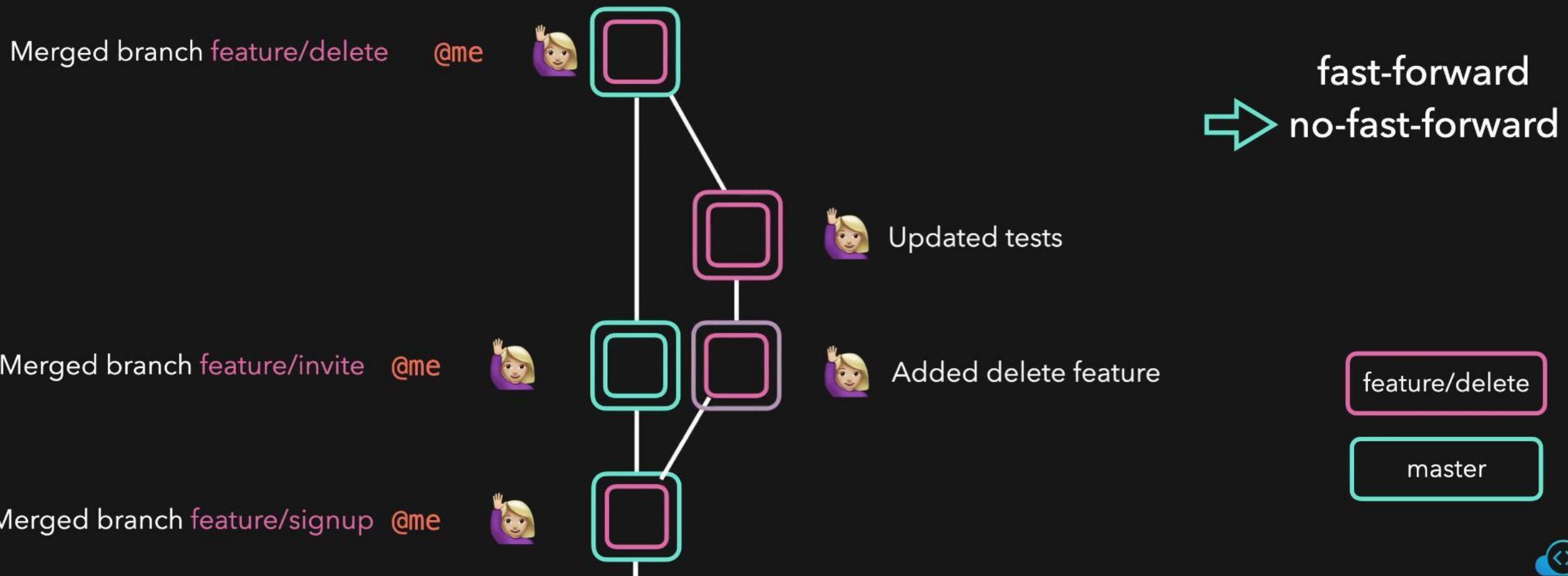


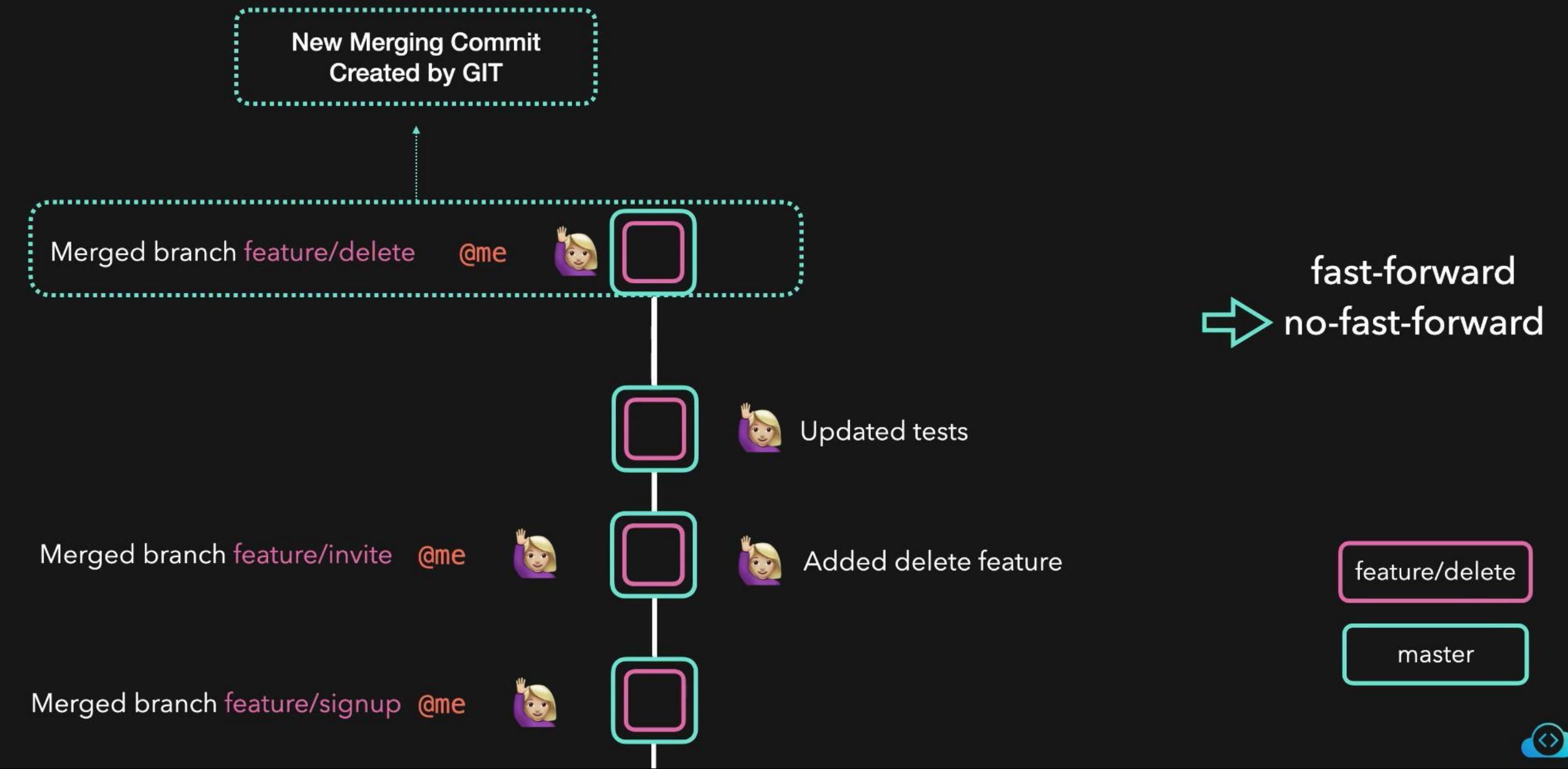
feature/signup

master



fast-forward
no-fast-forward

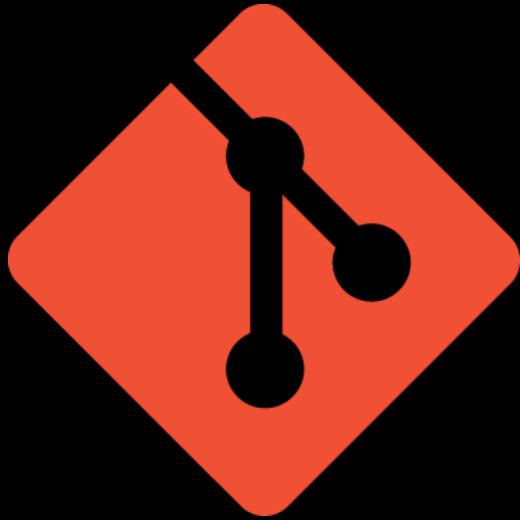






KodeKloud

Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>



Initializing a remote repo



GitHub



GitLab



Bitbucket



GitHub



GitLab



Bitbucket

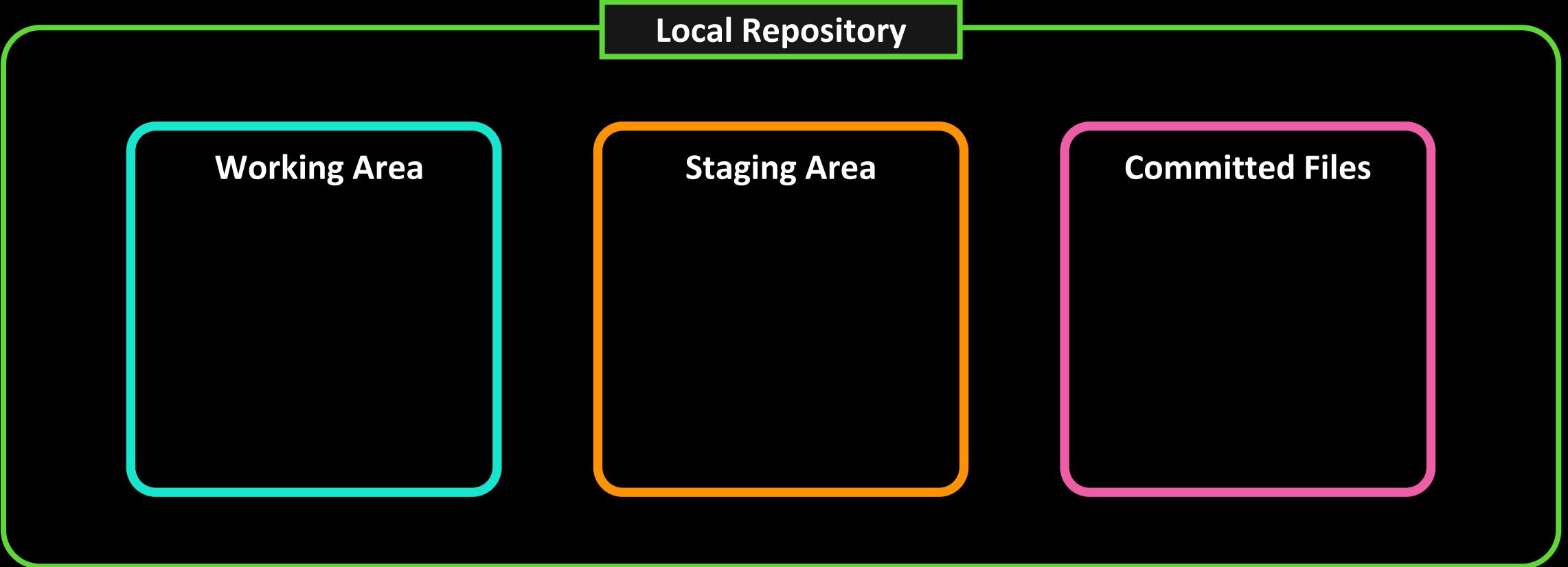
Connection
String

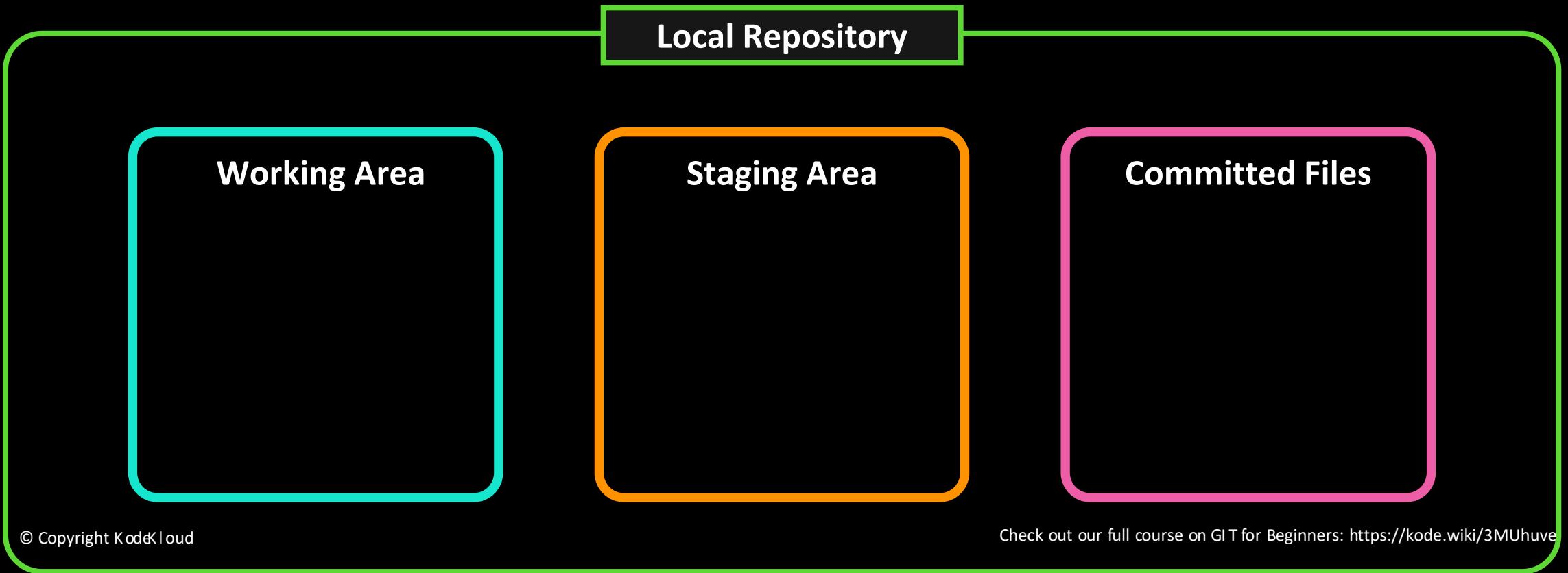
`https://.../.../[name].git`



bash

```
$ git remote add origin https://.../.../[name].git
```





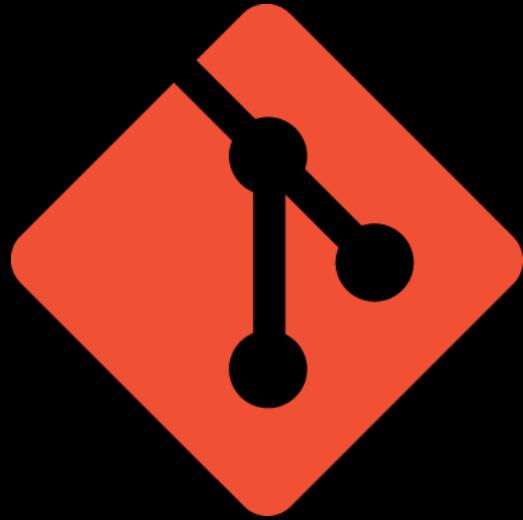


```
$ git remote -v  
origin  https://.../.../[name].git (fetch)  
origin  https://.../.../[name].git (push)
```



KodeKloud

Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>



Pushing

Remote Repository



Local Repository

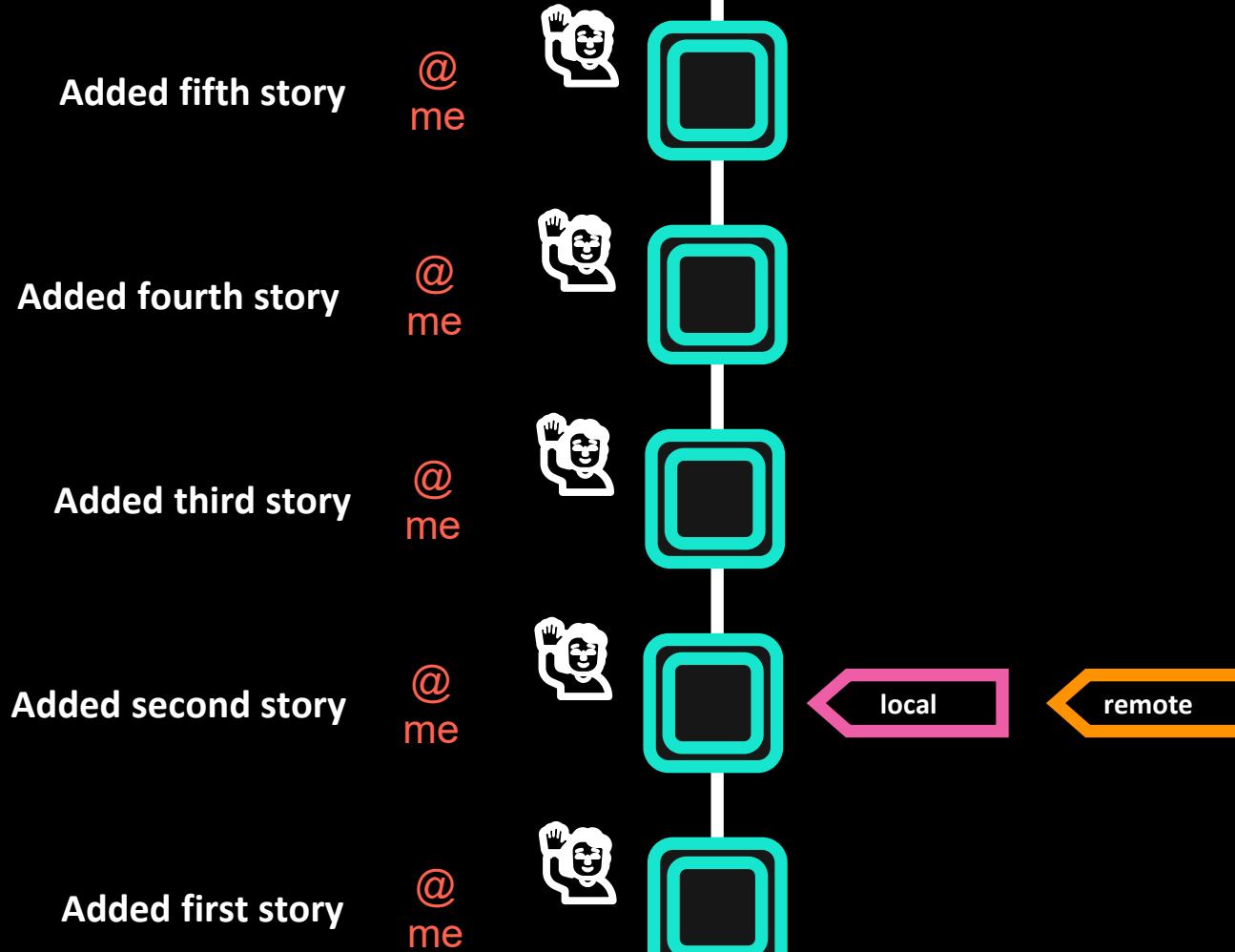
```
<html>
<body>
<h1>My Website!</h1>
</body>
</html>
```





bash

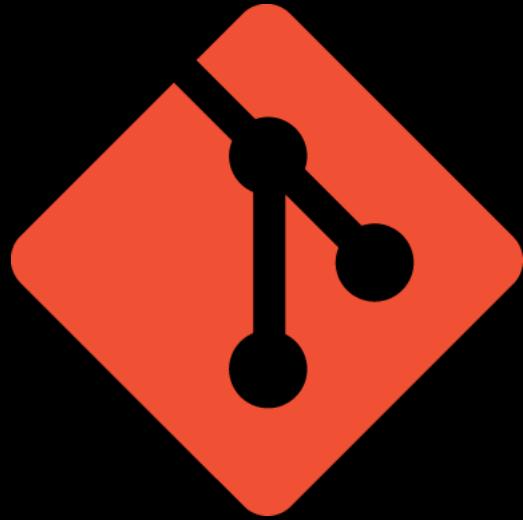
```
$ git push origin master
```





KodeKloud

Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>



Cloning



bash

```
$ git clone <ssh link>
```

 [lydiahallie / javascript-questions](#)

Watch 662 Star 20.8k Fork 2.6k

[Code](#) [Issues 35](#) [Pull requests 27](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [...](#)

Branch: master ▾

Go to file Add file ▾ Clone ▾

Clone with SSH ⓘ Use HTTPS
Use a password protected SSH key.
`git@github.com:lydiahallie/javascript-questions`

[Open in Desktop](#) [Download ZIP](#)

lydiahallie committed ac1fc96 11 days ago

 ar-AR	Update type list in question 34	3 months ago
 ar-EG	Typo correction of serialis	3 months ago
 bs-BS	Update README for other languages	3 months ago
 de-DE	Update type list in question 34	3 months ago

About 

A long list of (advanced) JavaScript questions, and their explanations ✨

[Readme](#) [MIT License](#)



bash

```
$ git clone git@github.com:.....git
```

```
Cloning into 'remote-repo'...
```

```
remote: Enumerating objects: 59, done.
```

```
remote: Counting objects: 100% (59/59), done.
```

```
remote: Compressing objects: 100% (43/43), done.
```

```
remote: Total 2948 (delta 28), reused 18 (delta 6)
```

```
Receiving objects: 100% (2948/2948), 1.93 MiB | 2.53 MiB/s, done.
```

```
Resolving deltas: 100% (1526/1526), done.
```



bash

```
$ cd remote-repo
```

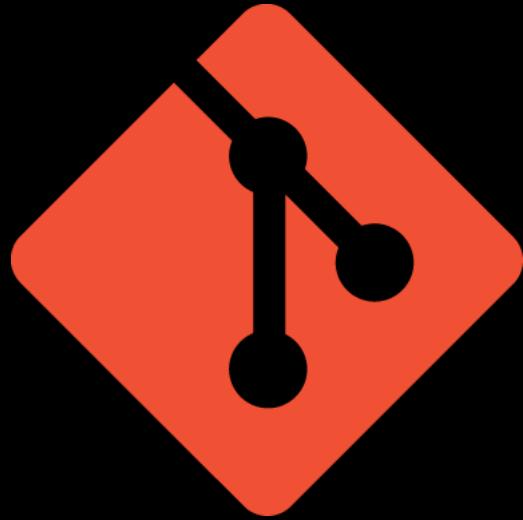
```
$ git log
```

```
commit 67c833e3...ecb7df62f (HEAD -> origin/master)
```

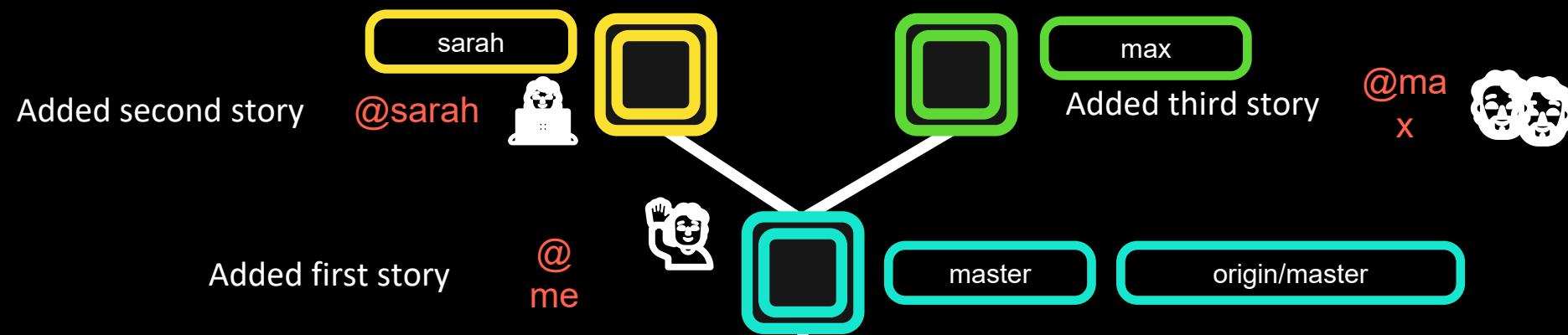
```
Author: John Doe <john@doe>
```

```
Date: Sun Jun 14 14:45:07 2020 -0700
```

Added first story



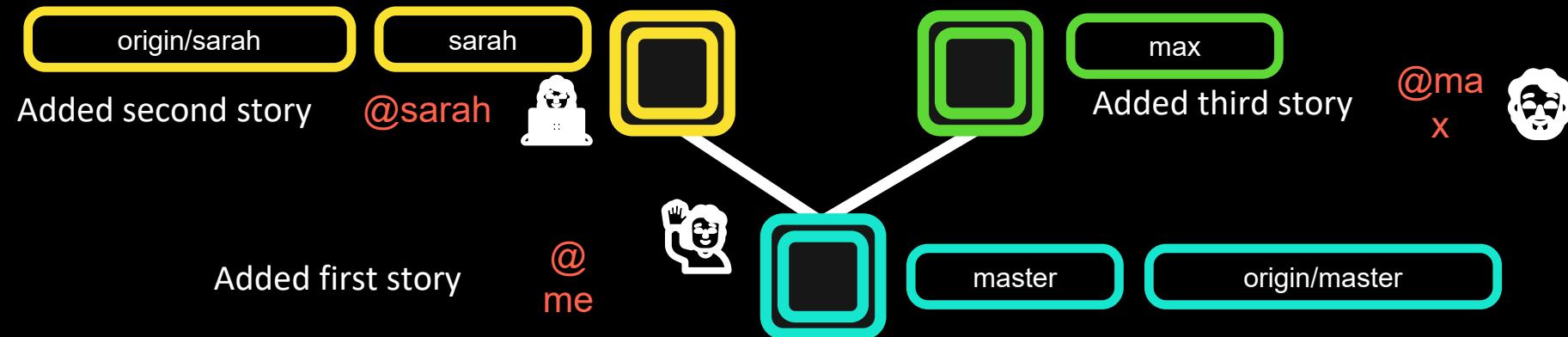
Pull Request





bash

```
$ git push origin sarah
```



master ▾

2 branches

0 tags

Go to file

Add file ▾

Code ▾

This branch is 1 commit behind sarah.

Pull request

Compare



lydiahallie Added first story

dc9ad3c 2 minutes ago 1 commits

first_story.md

Added first story

2 minutes ago

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).



base: master ▾



compare: sarah ▾

✓ Able to merge. These branches can be automatically merged.

Create pull request

Discuss and review the changes in this comparison with others.



-o 1 commit

± 1 file changed

0 commit comments

1 contributor



Commits on Jul 11, 2020



lydiahallie

Added second story

fb9f191

+ Showing 1 changed file with 1 addition and 0 deletions.

Unified Split

1 second_story.md

<> ...

```
... ... @@ -0,0 +1 @@  
1 + ## Second Story
```

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

Added second story #1

 Open

[lydiahallie](#) wants to merge 1 commit into [master](#) from [sarah](#) 

 Conversation 0

-o Commits 1

...

- Added a second story

 Added second story

fb9f191

Add more commits by pushing to the `sarah` branch on [lydiahallie/myrepo](#).



 This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



Write

Preview

H B T E <> C ii := ✓ @ ↵ ↶ ↷

[Leave a comment](#)

Attach files by dragging & dropping, selecting or pasting them.

 Close pull request

 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Added second story #1

Merged

lydiahallie merged 1 commit into master from sarah now

 Conversation 0

-o Commits 1

± Files changed 1

- Added a second story

- Added second story

fb9f191

lydiahallie merged commit [2c8f09e](#) into master now

Revert

Write

Preview

H

8

8

2

—

@

4

7

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Comment

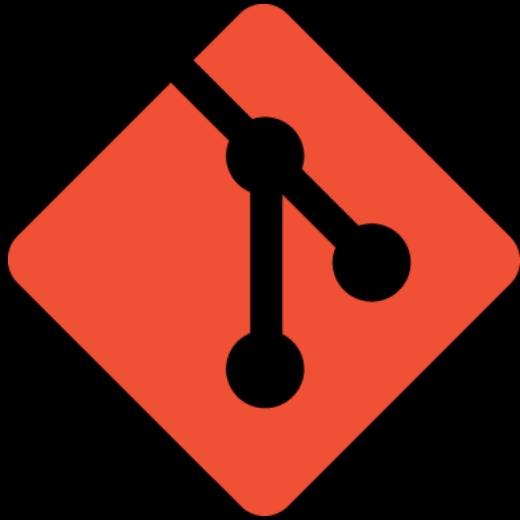
 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

 **ProTip!** Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.



KodeKloud

Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>

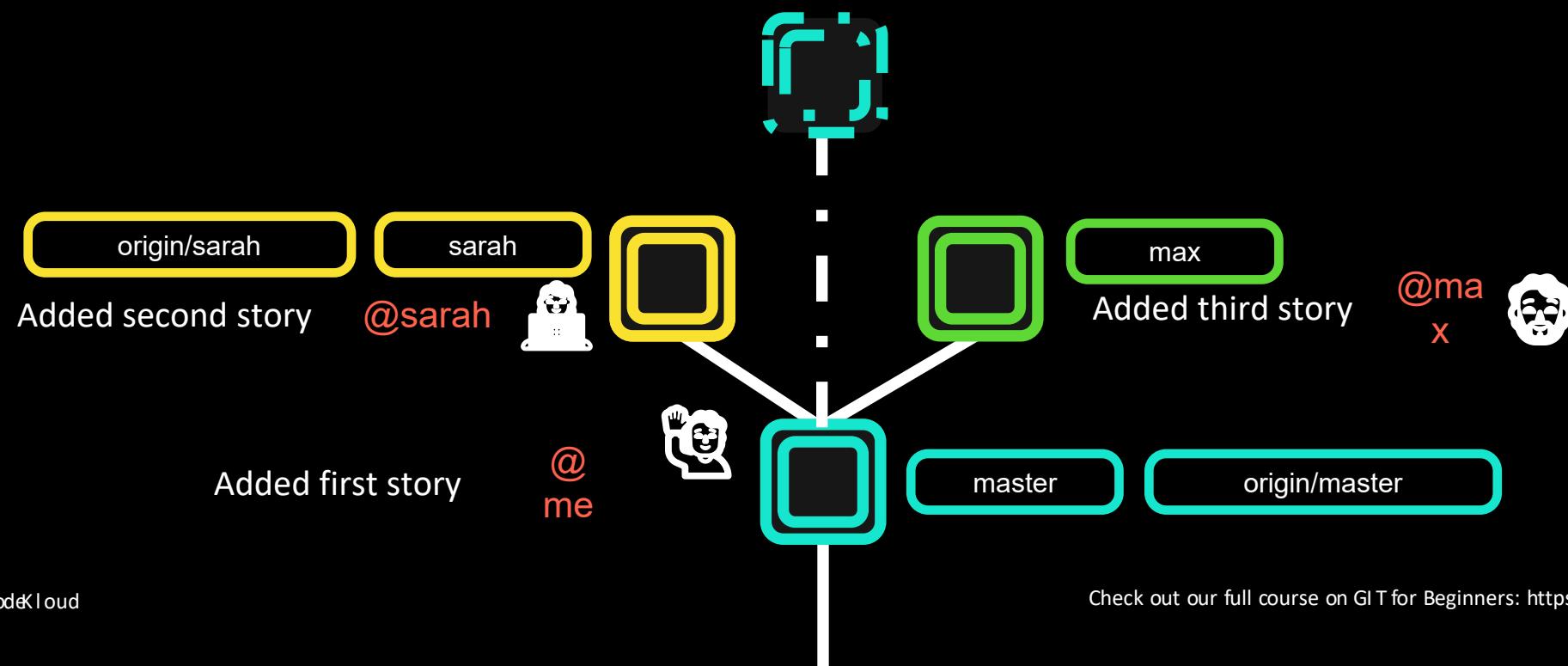


Fetching and Pulling



bash

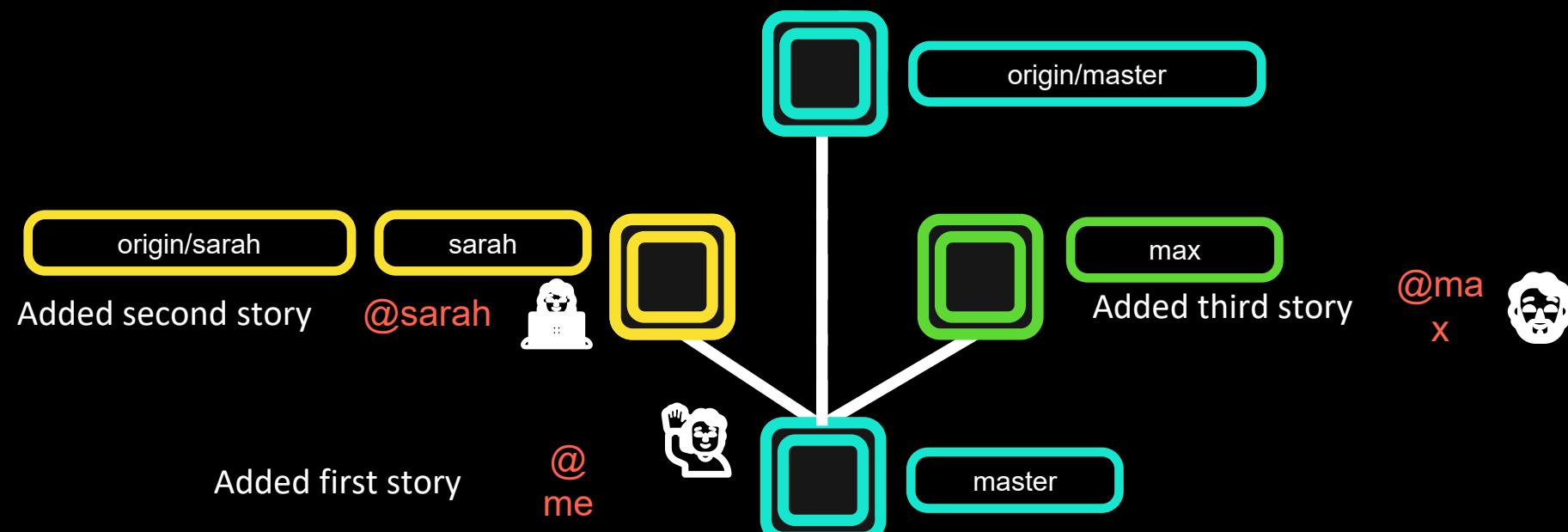
```
$ git fetch origin master
```





bash

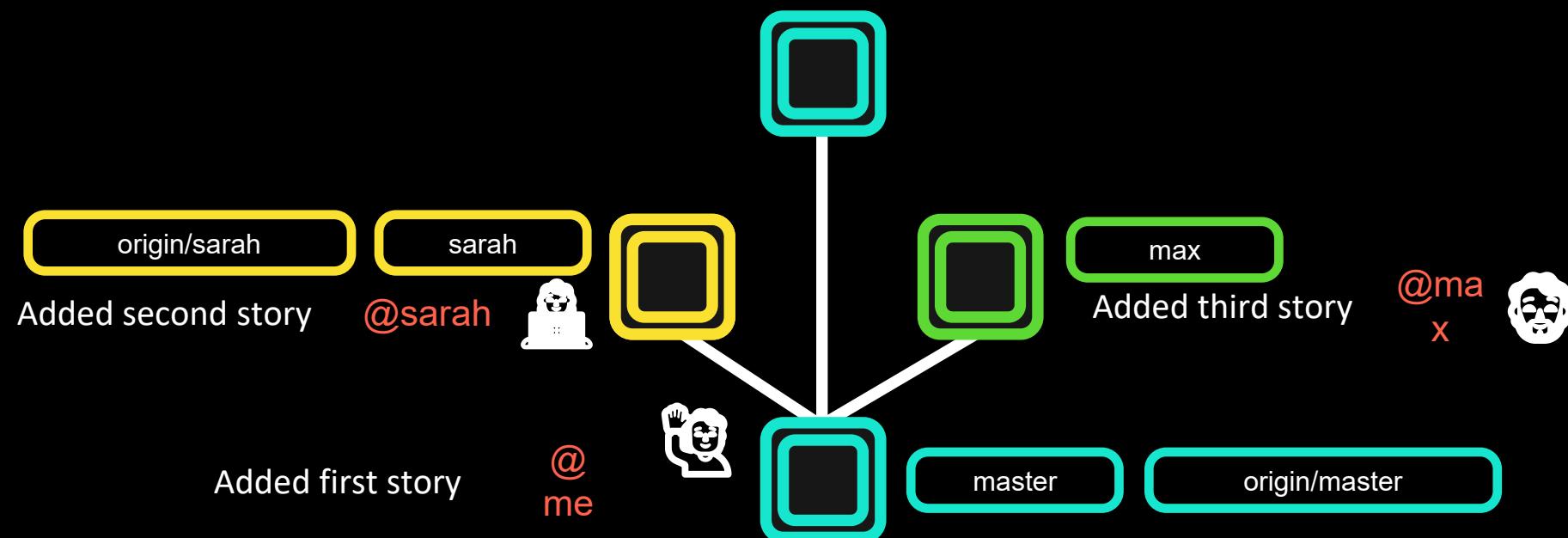
```
$ git merge origin/master
```





bash

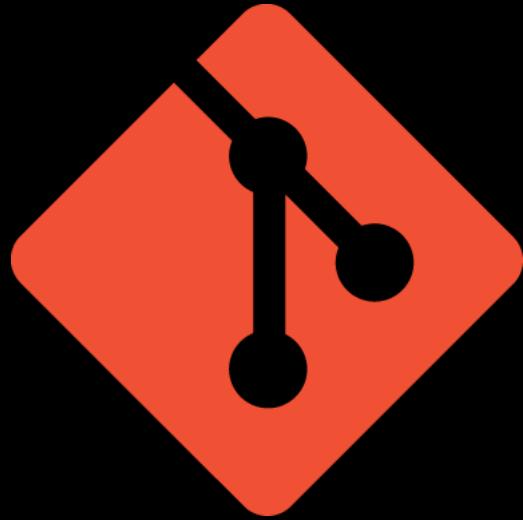
```
$ git pull origin master
```



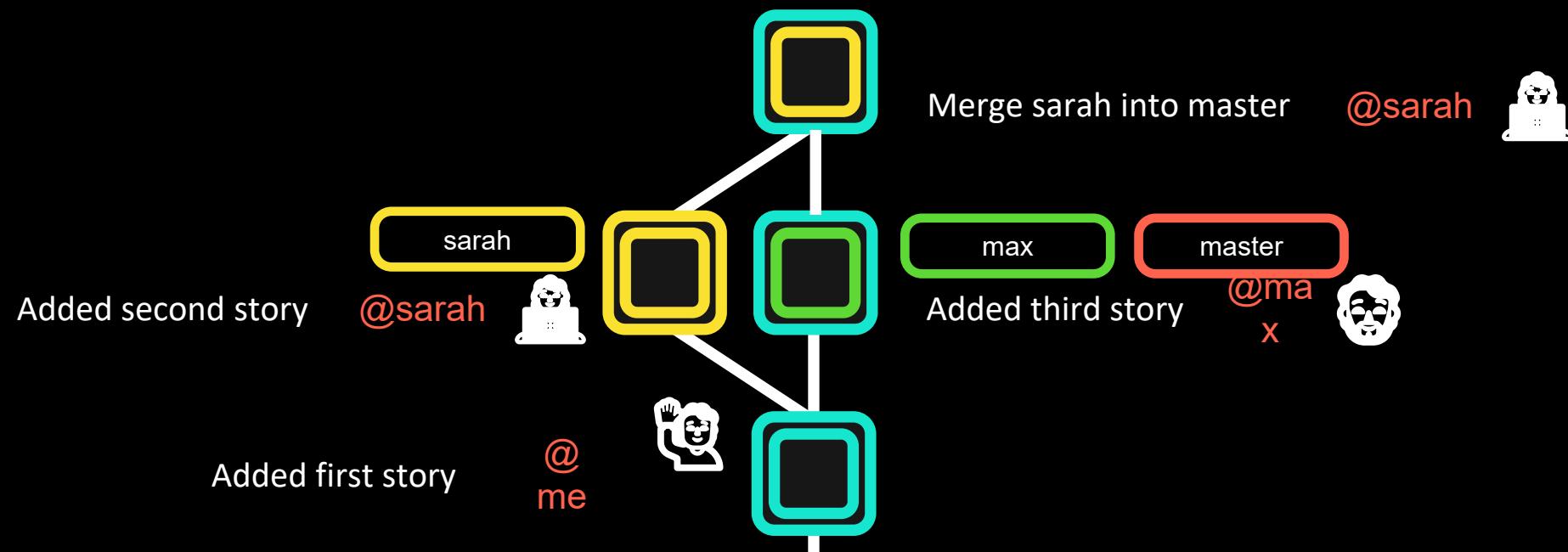


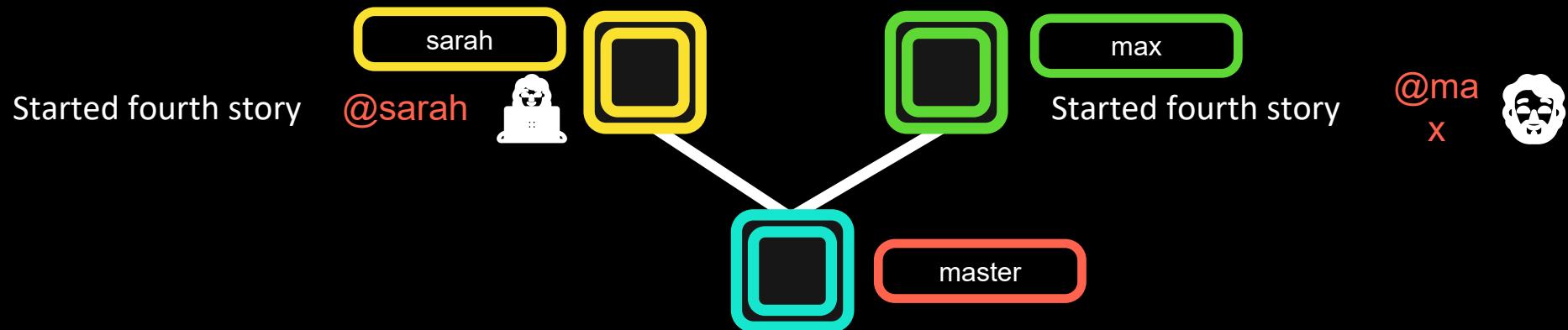
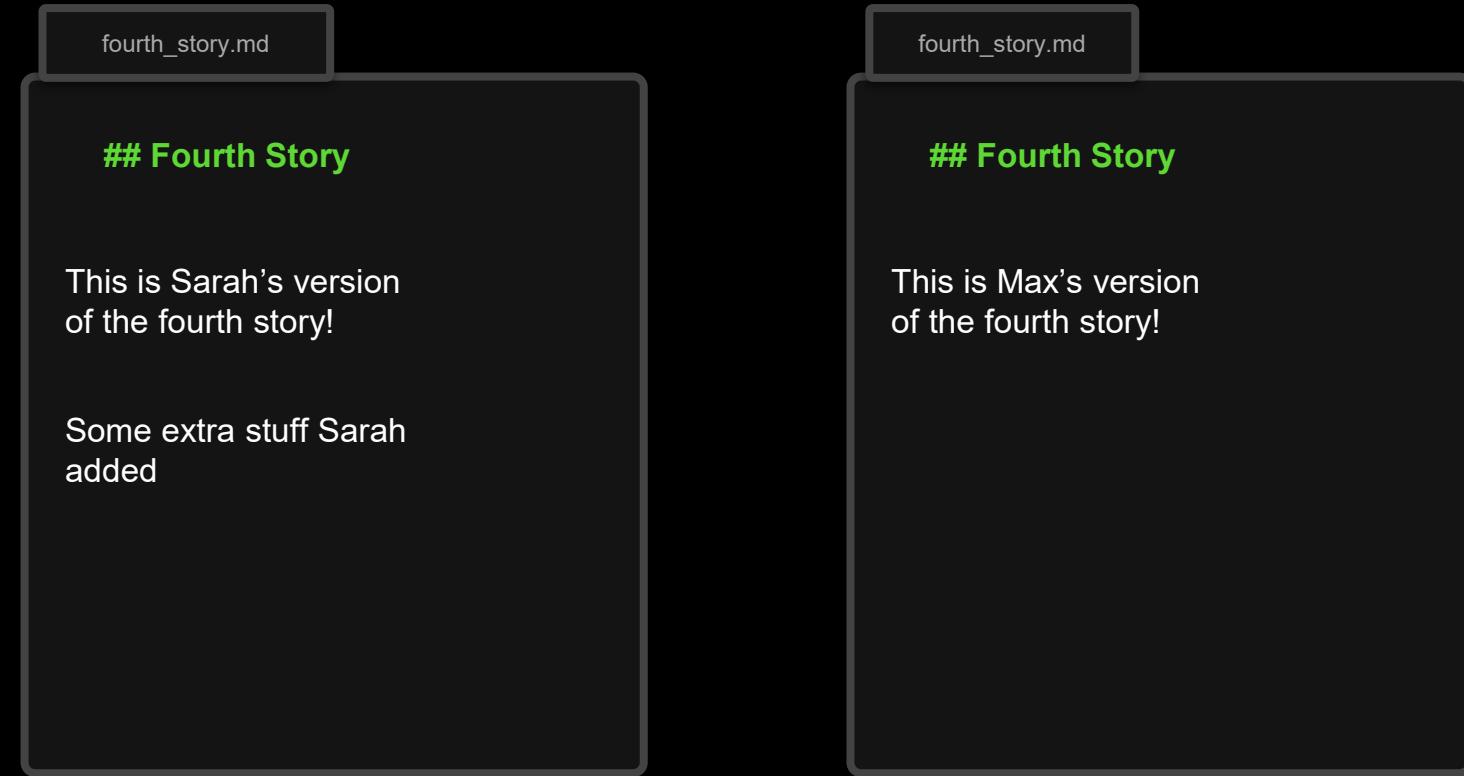
KodeKloud

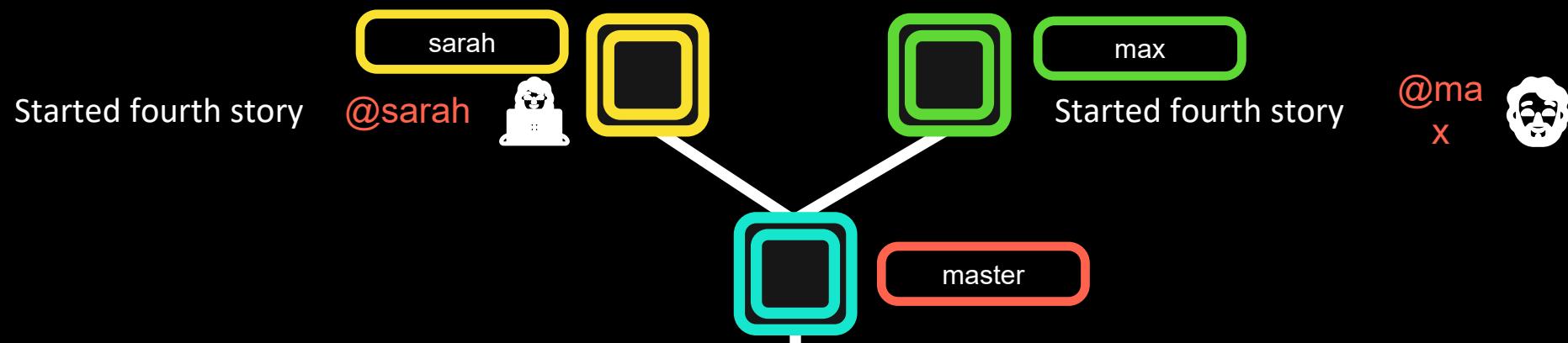
Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>

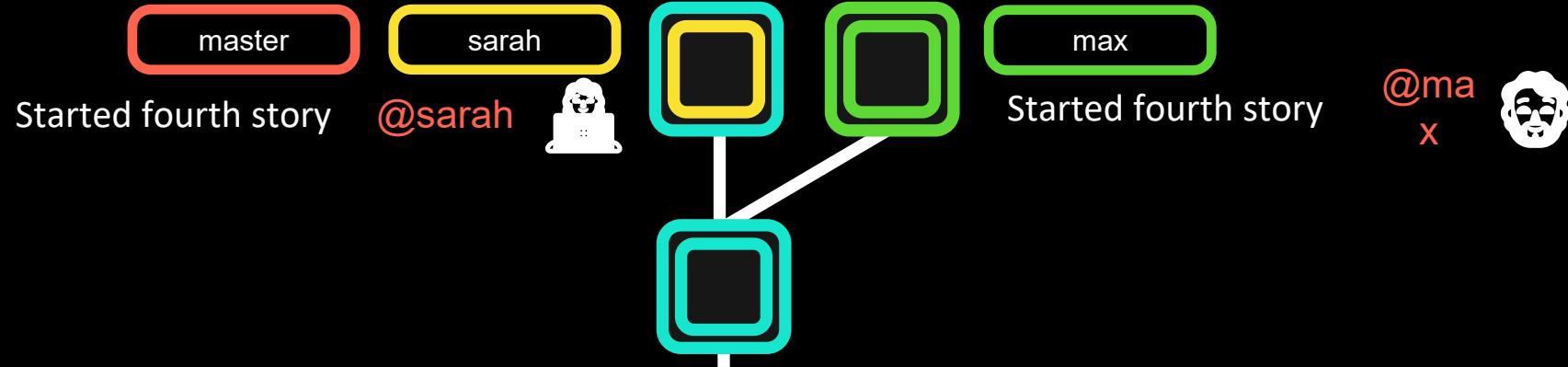


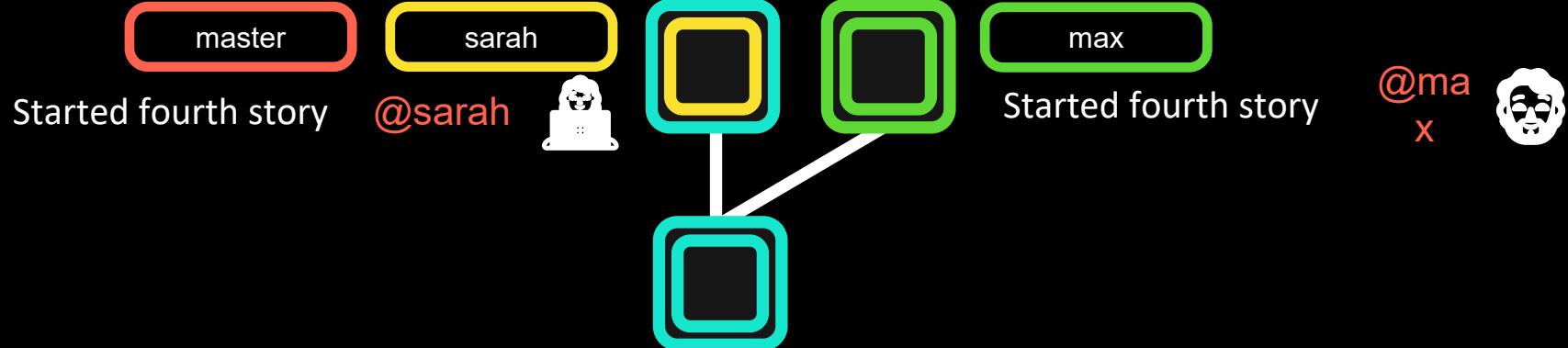
Merge Conflicts











Fourth Story

<<<<<< HEAD

This is Sarah's version

This is Max's version

>>>>> max

of the fourth story!

Some extra stuff Sarah
added

Fourth Story

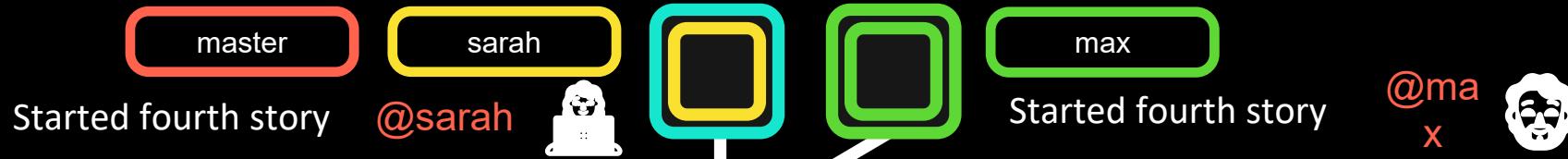
This is Max's version
of the fourth story!

Some extra stuff Sarah
added



bash

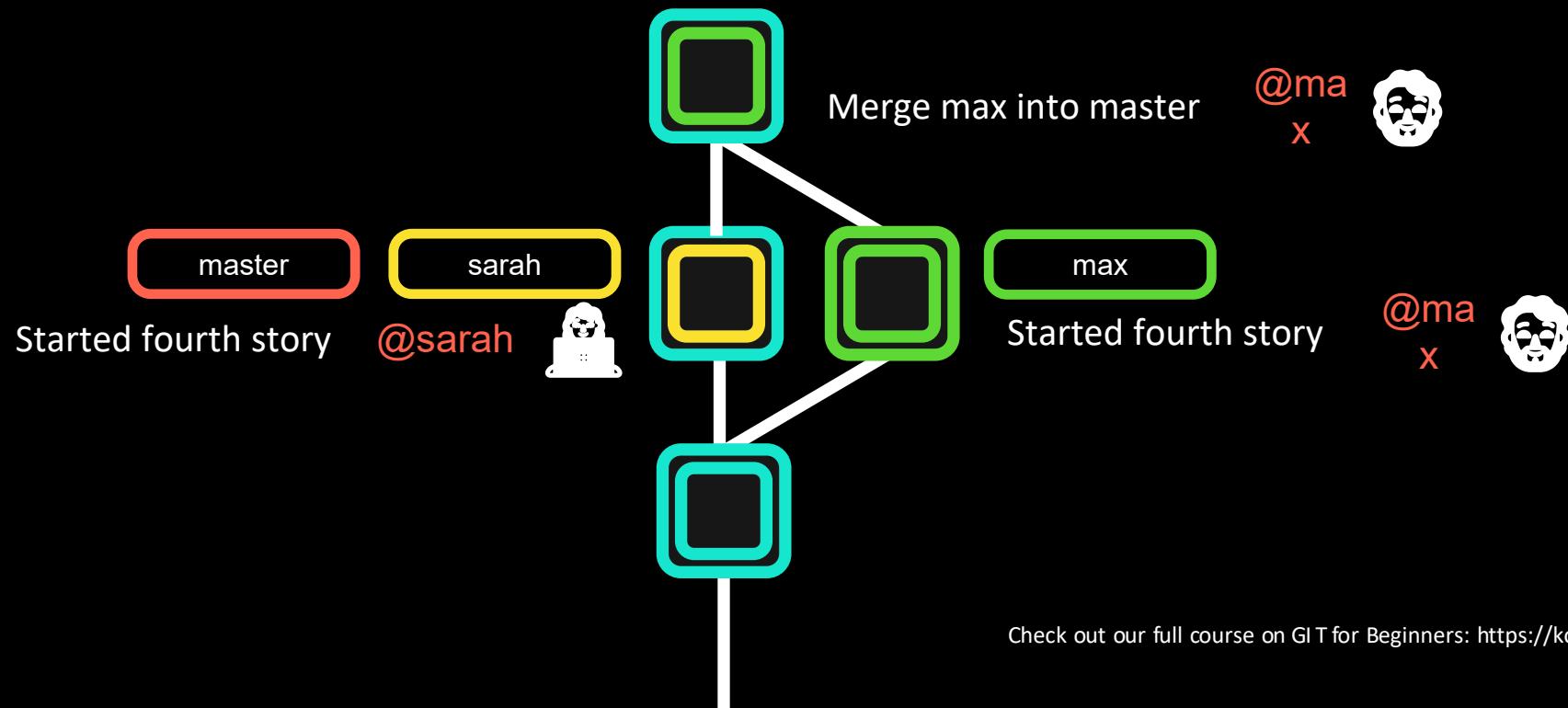
```
$ git add fourth_story.txt  
$ git merge
```





bash

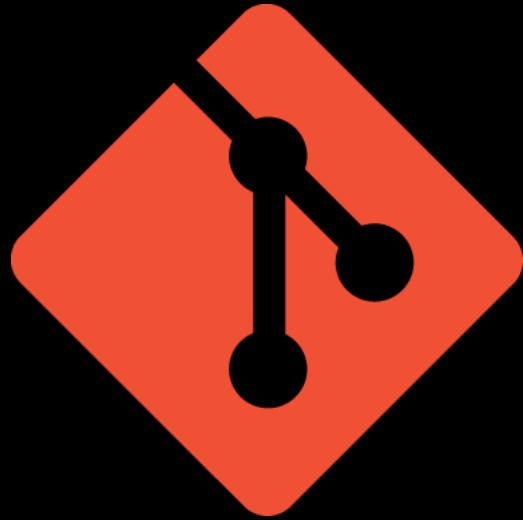
```
$ git add fourth_story.txt  
$ git merge
```





KodeKloud

Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>



Fork

other scheduling strategies, provides workflow to choreograph jobs to resolve dependencies, supports distributed computing of tasks, simple to use, powerful, well-documented, welcome you to access to use!

● Java ⭐ 1,693 📈 204 Built by 

⭐ 1,113 stars this month

❑ PaddlePaddle / PaddleOCR

Awesome OCR toolkits based on PaddlePaddle (8.6M ultra-lightweight pre-trained model, support training and deployment among server, mobile, embeded and IoT devices)

● C++ ⭐ 2,532 📈 456 Built by 

⭐ 1,779 stars this month

❑ iluwatar / java-design-patterns

Design patterns implemented in Java

● Java ⭐ 59,796 📈 19,133 Built by 

⭐ 1,321 stars this month

❑ taosdata / TDengine

An open-source big data platform designed and optimized for the Internet of Things (IoT).

● C ⭐ 12,622 📈 3,268 Built by 

⭐ 924 stars this month

❑ openai / gpt-2

Code for the paper "Language Models are Unsupervised Multitask Learners"

● Python ⭐ 12,838 📈 3,168 Built by 

⭐ 605 stars this month

❑ lyswhut / lx-music-desktop

一个基于 electron 的音乐软件

● Vue ⭐ 7,551 📈 1,299 Built by 

⭐ 1,129 stars this month

❑ TeamNewPipe / NewPipe

A libre lightweight streaming front-end for Android.

⭐ Star

Pull Request dave @friend



Added fifth story @me



Added fourth story @me



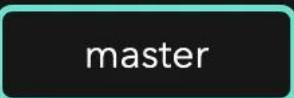
Added seventh story @friend



Added sixth story @friend



Fixed fourth story @friend





KodeKloud

Check out our full course on GIT for Beginners: <https://kode.wiki/3MUhuve>

What is Jenkins



Jenkins is one of the most popular automation tool used worldwide for continuous integration and continuous delivery.

Jenkins is a free and open-source automation server that enables developers to build, integrate, and test code automatically as soon as it is committed to the source repository.

Why Jenkins?

Challenges

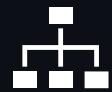
When working on a project with different teams, developers often face issues with different teams using different CI tools, version management, and other tools.

Setting up a CI/CD toolchain for each new project will lead to certain challenges like:

- Slower Releases
- Manual Builds
- Non-repeatable processes
- No Automations



Slower Releases



Manual Builds



Non-repeatable
processes



No automations

Why Jenkins?

Solution

Jenkins is the solution to those challenges.
It provides:

- Automated builds
- Automated Tests
- Automated CI/CD pipelines
- Automated deployments
- Ability to install Jenkins locally
- Jenkins support and Plugins



Automated builds



Automated Tests

CI

CD

Automated CI/CD
pipelines



Automated
deployments



Ability to install
Jenkins locally



Jenkins support
and Plugins



KodeKloud

Check out our full course on Jenkins: <https://kode.wiki/3oQvNJn>

Why Jenkins?



Open-source



1000+ plugins



Free



Paid, Enterprise

Free!

Jenkins is free and you don't have to pay for anything.
Jenkins can be hosted on a Virtual Machine, a container. Or
even locally for development purposes.



Plugins

Jenkins is well tested and provide several integrations with 1800+ plugins to support build, deployment and automation for the project



The screenshot shows the Jenkins Plugins Index page. At the top left is a circular icon featuring a cartoon character wearing a hard hat and safety vest, surrounded by various icons like a wrench, a gear, and a shield. To the right of the icon, the title "Plugins Index" is displayed in a large, bold, white font. Below the title, a subtitle reads "Discover the 1800+ community contributed Jenkins plugins to support building, deploying and automating any project." A search bar with the placeholder "Find plugins..." and a magnifying glass icon is positioned below the subtitle. On the left side of the main content area, there's a sidebar titled "Browse categories" with links to "Platforms", "User interface", "Administration", "Build management", and "Source Code Management". The main content area is divided into four sections: "New Plugins", "Recently updated", and "Trending". Each section lists several plugin names.

New Plugins	Recently updated	Trending
CloudBees Feature Management	MSTestRunner	Localization: Chinese (Simplified)
Artifactz.io	Deployed On Column	Pipeline: GitHub Groovy Libraries
Jakarta Activation API	InfluxDB	Timestamper
Jakarta Mail API	MATLAB	Infrastructure for Publish Over X
JavaMail API	Cloudify	Gradle
JavaBeans Activation Framework (JAF) API	Allure	Role-based Authorization Strategy
Jersey 2 API	InsightVM Container Image Scanner	GitLab
Shutdown Queue	Build Monitor View	Build Timeout
BMC DevOps for CFA	URLTrigger	NodeJS
XTrigger API	XTrigger API	Workspace Cleanup

Enterprise Options



Support



Managed
Service

The screenshot shows the CloudBees homepage. At the top, there's a navigation bar with the CloudBees logo and links for Platform, Use Cases, Customer Stories, Resources, Developers, and Why CloudBees. Below the navigation, a banner features the text "CloudBees Ranked #11 in G2 Top 50 Enterprise Software P" and "Continuous Integration for the Enterprise". The main text below the banner reads: "Built on the most widely used automation server in the world Jenkins™ - CloudBees Continuous Integration provides flexible, scalable and governed CI/CD you can trust." At the bottom of the page are two buttons: "Schedule a Demo" and "Contact Sales".



KodeKloud

Check out our full course on Jenkins: <https://kode.wiki/3oQvNJn>

Continuous Integration

Continuous Integration is a process in which the code is merged from multiple contributors and added to a single repository.

In simple words, CI is a process to take the code package it and send it to the CD for further processing.

Continuous Deployment

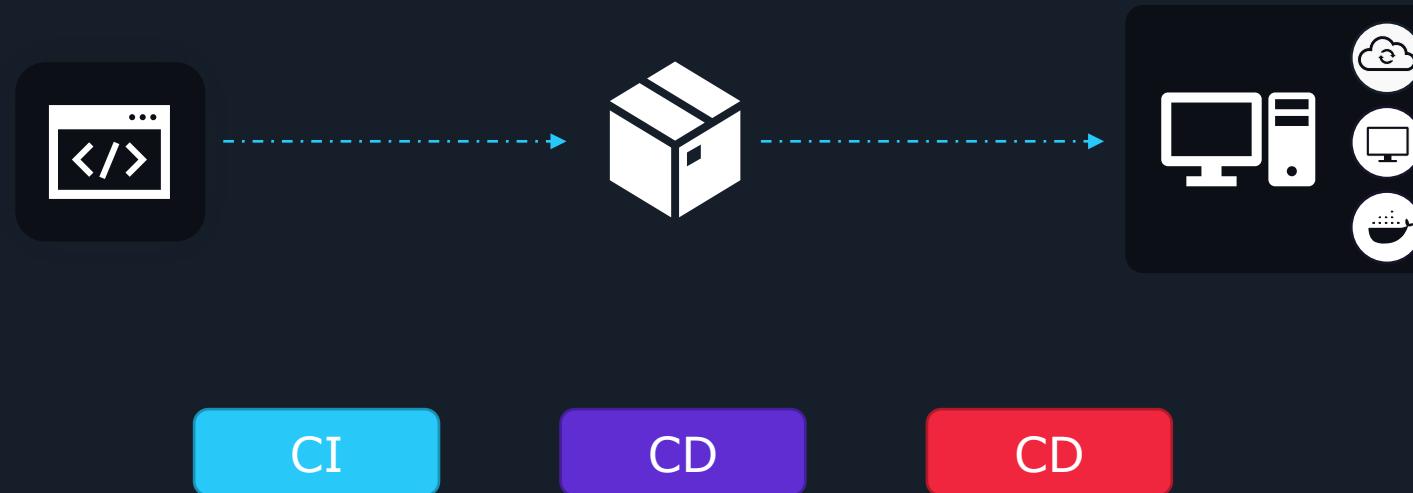
Continuous Deployment is an automated process in which the code is taken from the repository and deployed to the system.

Continuous Integration and Continuous Delivery/Deployment (CICD)

CI/CD in simple words is a process to take a code, package it up and deploy it to a system that can be serverless, a VM, or a container.

CI/CD can be broken down into 3 steps:

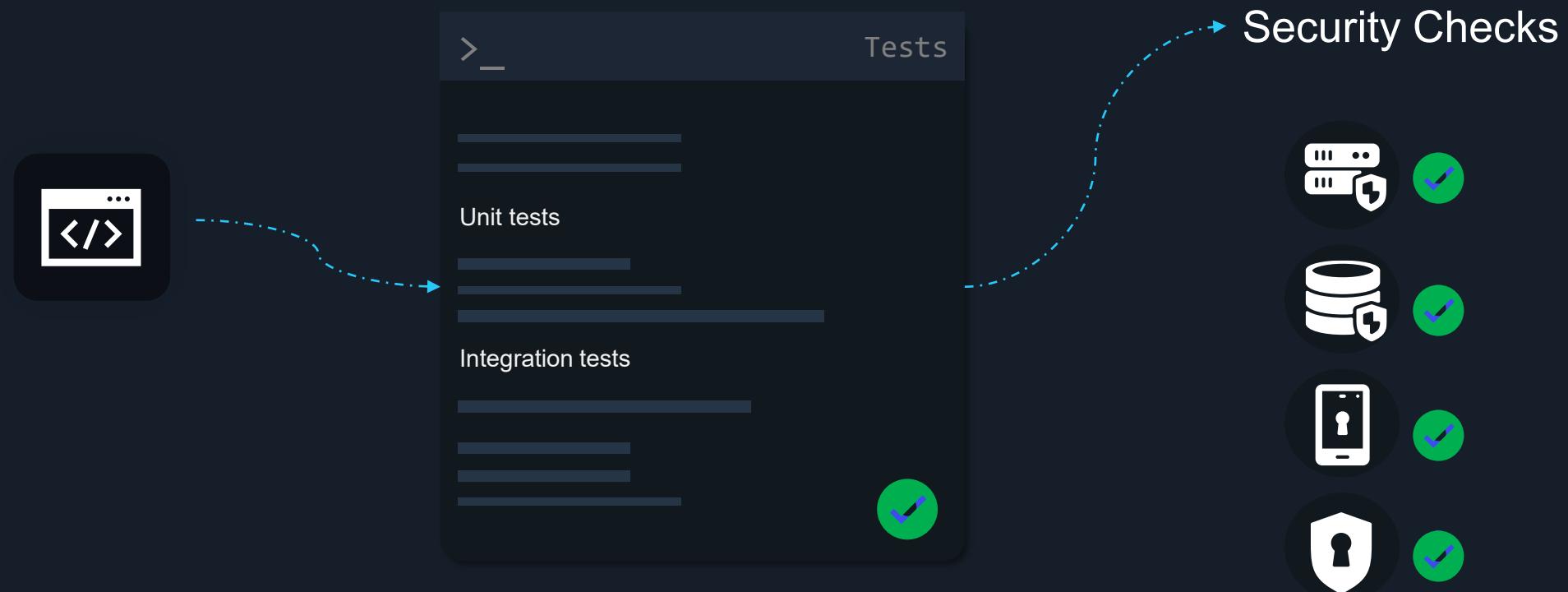
- CI – Continuous Integration
- CD – Continuous Delivery
- CD – Continuous Deployment



The Key Pieces of CI

Key Processes of Continuous Integration

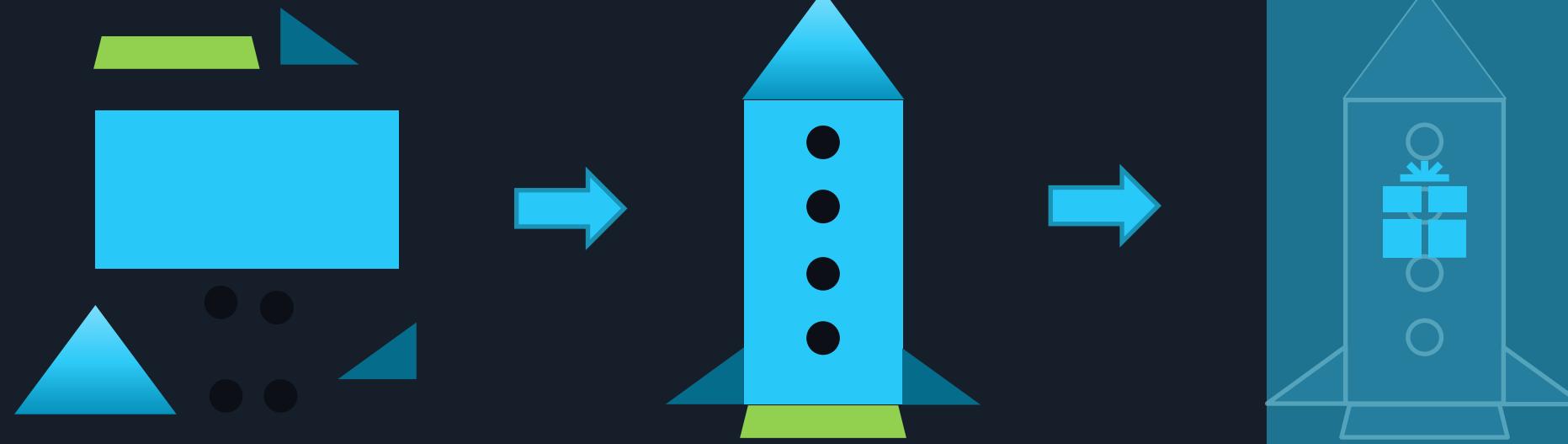
- Package up the code
- Test the code (run unit tests, integration tests, etc)
- Run security checks against the code



Continuous Integration (CI)

Think of the Continuous Integration process like a gift you're wrapping

- The gift comes in pieces
- You put the gift together (maybe a toy chest/box)
- The gift gets wrapped in wrapping paper
- You put it in the car and deliver it to the person.

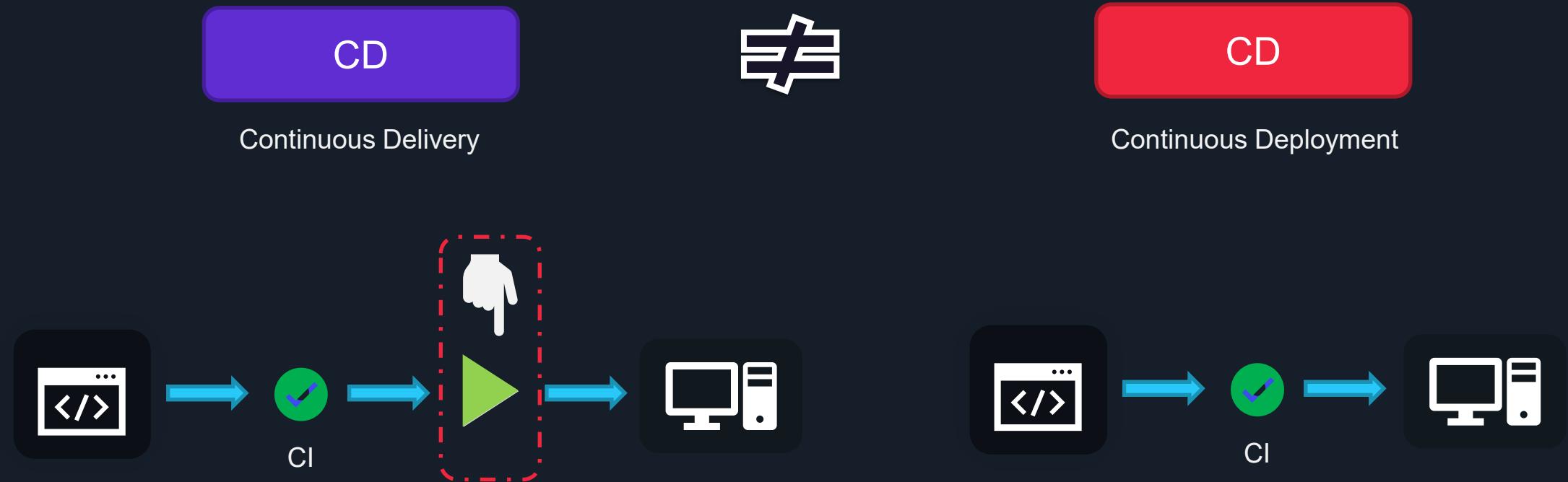


Continuous Integration (CI)



Continuous Deployment vs Continuous Delivery

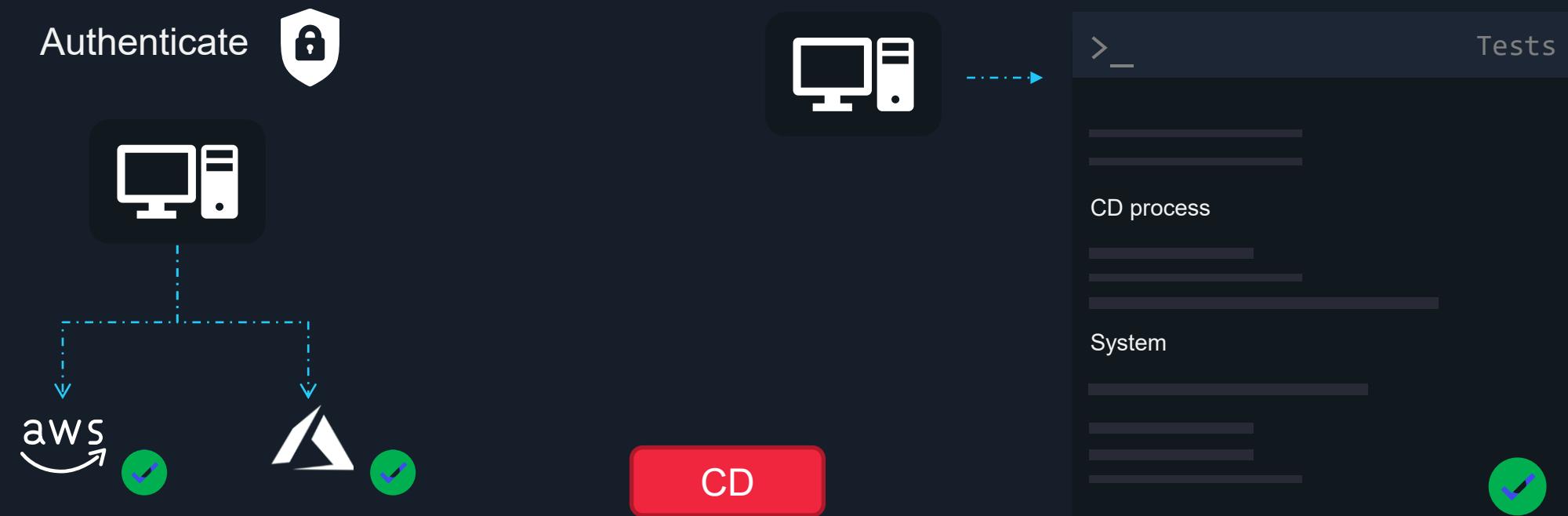
The basic difference between Continuous Delivery and Continuous Deployment is that in Continuous Delivery to deploy the code after the CI process you have to manually trigger it via some button to deploy on the system whereas in Continuous Deployment this process is automatic.



Key Pieces of CD

Key Pieces of CD:

- Ensure you're authenticated to the system or wherever you're deploying
 - Ensure that the code that's being deployed is working as expected once it's deployed





KodeKloud

Check out our full course on Jenkins: <https://kode.wiki/3oQvNJn>

Installing Jenkins



<https://www.jenkins.io/doc/book/installing/linux/>

Install Jenkins on Ubuntu

Step 1: Install Java on Ubuntu

```
$ sudo apt update  
$ sudo apt install openjdk-8-jdk
```

Alternatively, install version 11:

```
$ sudo apt install openjdk-11-jdk
```

Confirm the download by pressing Y and Enter

Step 2: Add the repository key to the system:

```
$ sudo apt wget -q -O -  
https://pkg.jenkins.io/debian-  
stable/jenkins.io.key | sudo apt-key add -
```

Step 3: Once the key is added with no errors, append the Debian package repository address

```
$ sudo sh -c 'echo deb  
http://pkg.jenkins.io/debian-stable  
binary/ >  
/etc/apt/sources.list.d/jenkins.list'
```

Step 4: Run update to use new repository

```
$ sudo apt update
```

Step 5: Install Jenkins

```
$ sudo apt install jenkins
```

Start Jenkins on Ubuntu

After successful installation let us start Jenkins

```
$ sudo systemctl start jenkins
```

The above command will not display any output

To check the running status of Jenkins use the below command which should show active status on run

```
$ sudo systemctl status jenkins
```



KodeKloud

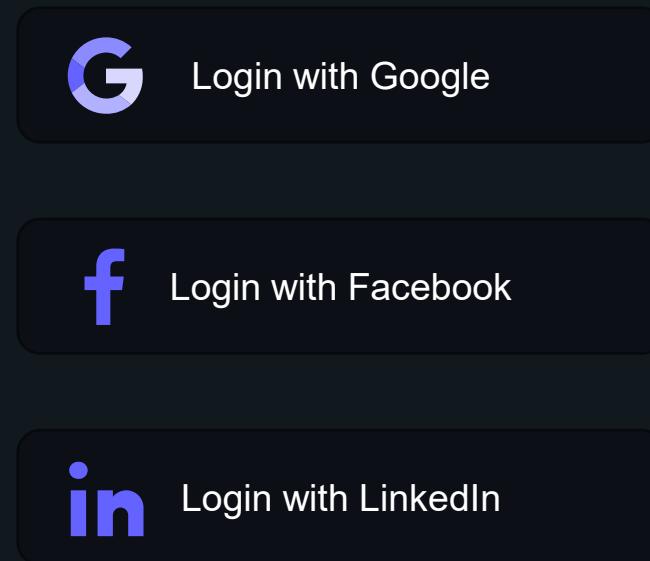
Check out our full course on Jenkins: <https://kode.wiki/3oQvNJn>

Jenkins Plugins

Plugins are used in Jenkins to enhance Jenkins functionality and cater to user-specific needs. Just like how Gmail, Facebook and LinkedIn help you connect your one service to another, plugins also work the same way and allow us to connect one service to other services and work with other products.



User



Website

Plugins

For example, you want to connect to Azure from Jenkins you would need to download Azure Plugin which will allow you to connect to Azure at a programmatic level.
Similarly, we can have other integrations with AWS, GitHub, etc using plugins.



Azure Plugin



AWS Plugin



Github Actions Plugin



Install Plugins

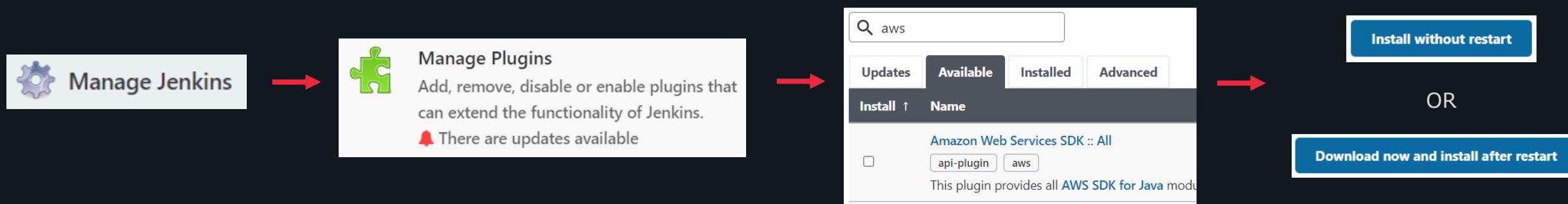
To install a new plugin in Jenkins

- 1) Go to Manage Jenkins -> Manager Plugins
- 2) Click Available and search for the desired plugin.
- 3) Select the desired plugin and Install.

Note: Few plugins may need a restart

To restart Jenkins

```
$ sudo systemctl restart jenkins
```



Update Plugins

To update any existing plugin in Jenkins

- 1) Go to Manage Jenkins -> Manager Plugins
- 2) Click Updates and search for the desired plugin.
- 3) Select the desired plugin and Install.

Note: Few plugins may need a restart

To restart Jenkins

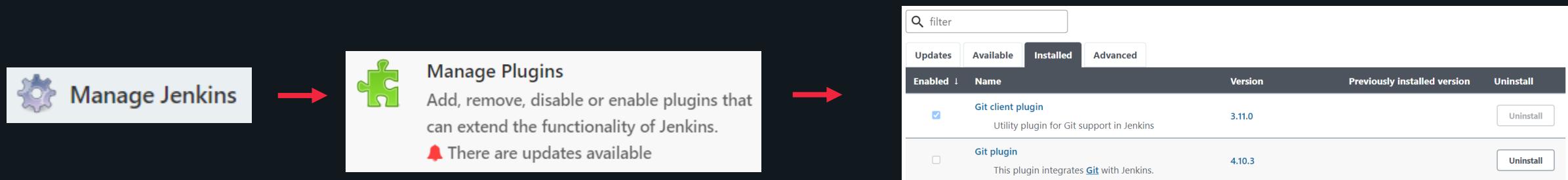
```
$ sudo systemctl restart jenkins
```



Delete Plugins

To delete any plugin in Jenkins

- 1) Go to Manage Jenkins -> Manager Plugins
- 2) Go to Installed and search for the desired plugin.
- 3) Click on uninstall button for the plugin you want to delete.
Click yes to proceed with the deletion.





KodeKloud

Check out our full course on Jenkins: <https://kode.wiki/3oQvNJn>

Jenkins Visuals



Visuals

Jenkins
Menu

Build

User
account

Navigate

Jenkins Jobs

Different types of jobs that can be created in Jenkins:

1) Freestyle project

This is a central feature of Jenkins. It will build the project, combine SCM with the build system. It can also be used for things other than building applications.

2) Pipeline

This is used to create a pipeline

3) Multi-configuration project

This is great if you need a large number of Jenkins configurations if you need multiple environments like Dev/ UAT.

4) Folder

This creates containers and stores nested items. It is useful in grouping, creating a namespace, etc.

5) Organisation folder

Creates a multibranch project for all different subfolders that are available.

6) Multibranch Pipeline

It sets up pipeline projects for different repositories.

Administering Jenkins



Backup



Restore



Monitor



Scale



Manage

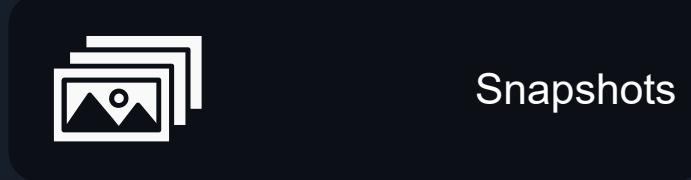
Backup and Restore



Backup

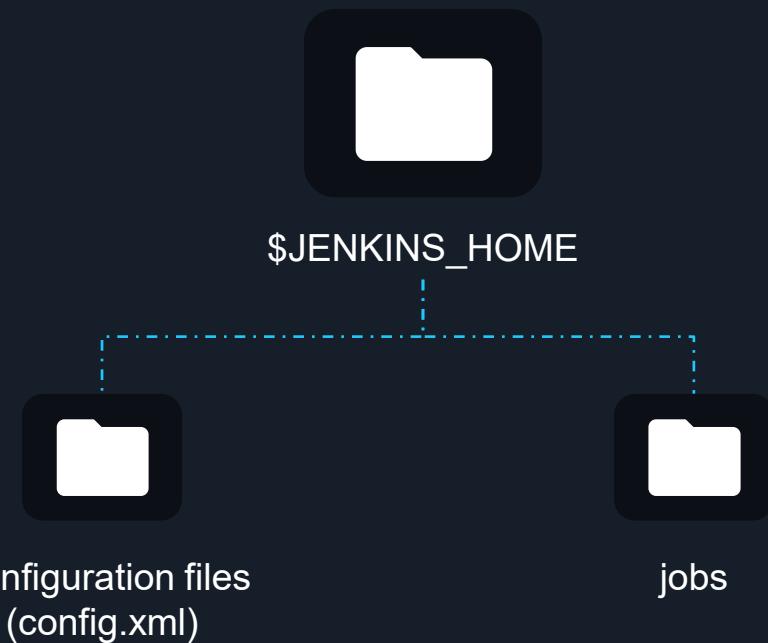


Full



Snapshots

Which Files To Backup?



Backup Jenkins

It is crucial to have adequate backups of your Jenkins instance. Backups are used to recover from accidental configuration changes. Recovering a file that has been mistakenly erased or has been corrupted. Or just to recover a previous setup.

There are two ways we can backup Jenkins:

- 1) Using Plugins
- 2) Using custom shell script

Backup Jenkins

To backup Jenkins using a plugin, you will first need to install a backup plugin. Some of the most commonly used plugins are ThinBackup, Periodic Backup, Google cloud Backup.

For backing up using any of these plugins there are a few general steps that must be followed:

- 1) Creating a backup directory with read and write access
- 2) Selecting files that need backup

Backing up using shell script

Please check out these popular repositories for your reference:

- 1) repository: <https://github.com/sue445/jenkins-backup-script>
- 2) gist: <https://gist.github.com/abayer/527063a4519f205efc74>



KodeKloud

Check out our full course on Jenkins: <https://kode.wiki/3oQvNJn>

Jenkinsfile

Jenkinsfile is a text file that contains definitions. This could be templates or instructions. It tells pipelines what they should be doing and what services and plugins they should be interacting with.

Components of Jenkinsfile:

- 1) Pipeline – The task you are trying to accomplish
- 2) Build Agent –The place where you run your pipeline
- 3) Stages – Staging/Production/UAT
- 4) Steps –Work done in the pipeline

What Is A Jenkinsfile?



Instructions

Templates

Jenkinsfile

What Is A Jenkinsfile?



Components of Jenkinsfile

The task you are trying to accomplish

Build Agent

Stages

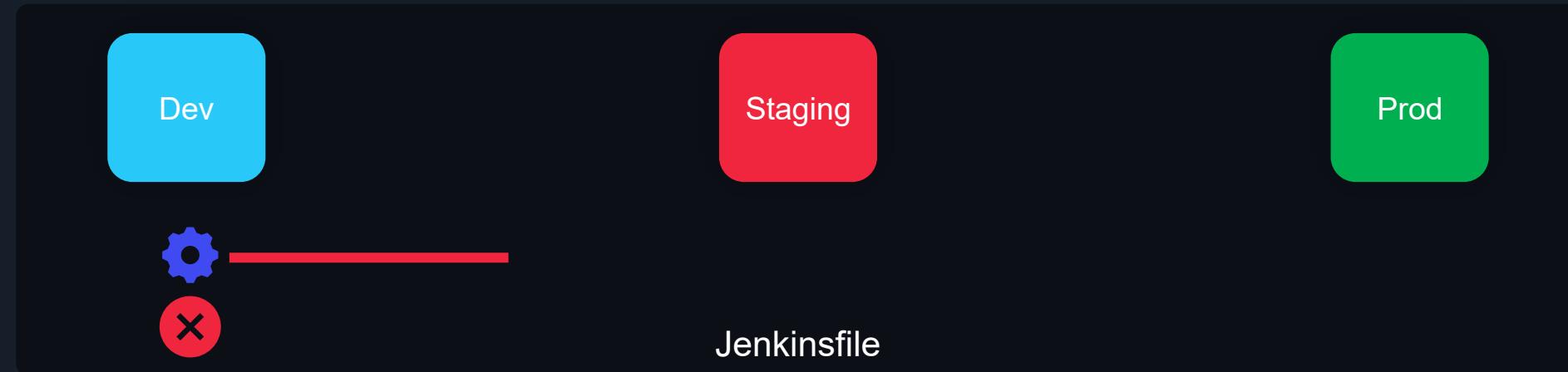
Steps

Jenkinsfile

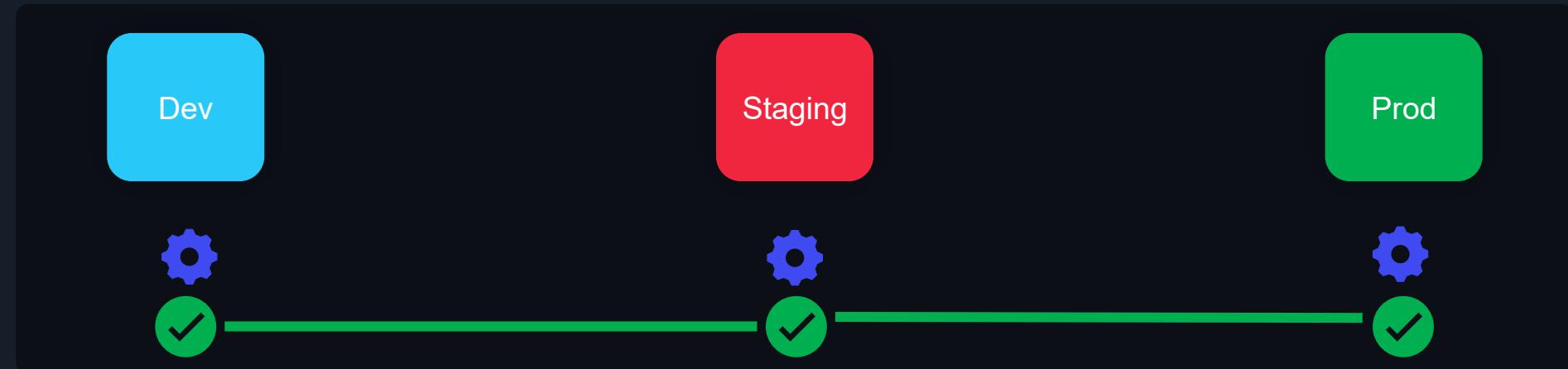
```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                echo 'Building..'
            }
        }
        stage('Test') {
            steps {
                echo 'Testing..'
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying....'
            }
        }
    }
}
```

Detailed description: This diagram illustrates the hierarchical structure of a Jenkinsfile. On the left, four components are listed: 'The task you are trying to accomplish', 'Build Agent', 'Stages', and 'Steps'. Red dashed arrows point from each of these labels to specific sections in a Jenkinsfile on the right. The 'The task you are trying to accomplish' arrow points to the top-level 'pipeline' block. The 'Build Agent' arrow points to the 'agent any' line within the 'pipeline' block. The 'Stages' arrow points to the 'stages' block. The 'Steps' arrow points to the 'steps' block within one of the 'stage' blocks.



Multi-Stage Pipelines



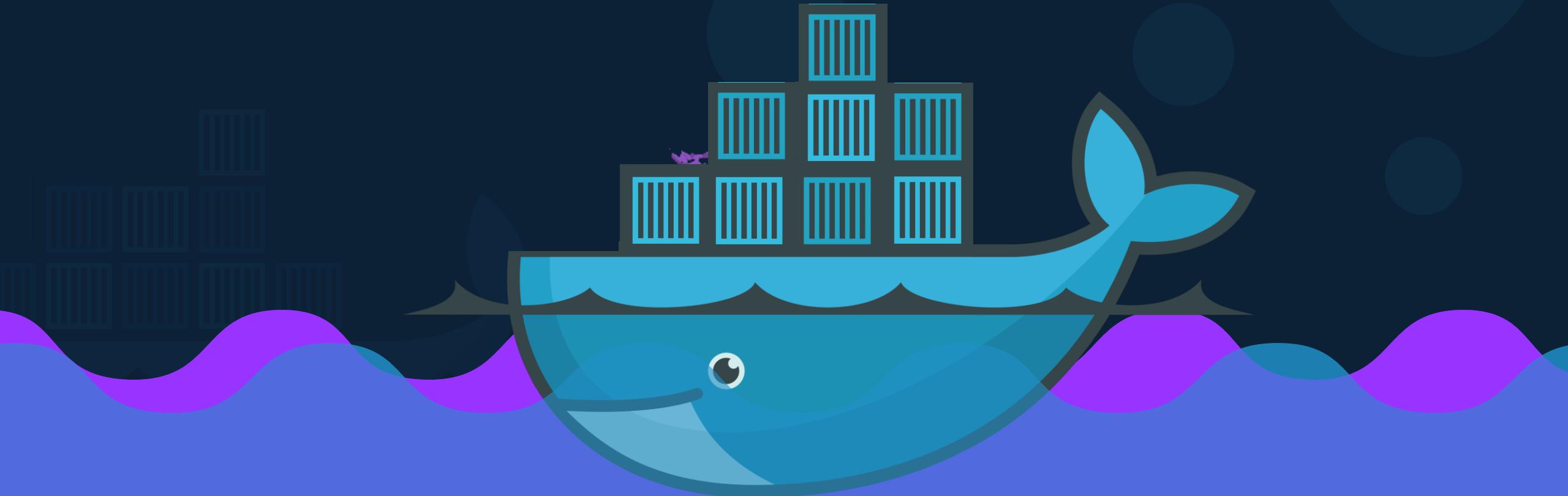


KodeKloud

Check out our full course on Jenkins: <https://kode.wiki/3oQvNJn>

docker

for beginners



Objectives

- What are Containers?
- What is Docker?
- Why do you need it?
- What can it do?

- Run Docker Containers
- Create a Docker Image
- Networks in Docker
- Docker Compose

- Docker Concepts in Depth

- Docker for Windows/Mac

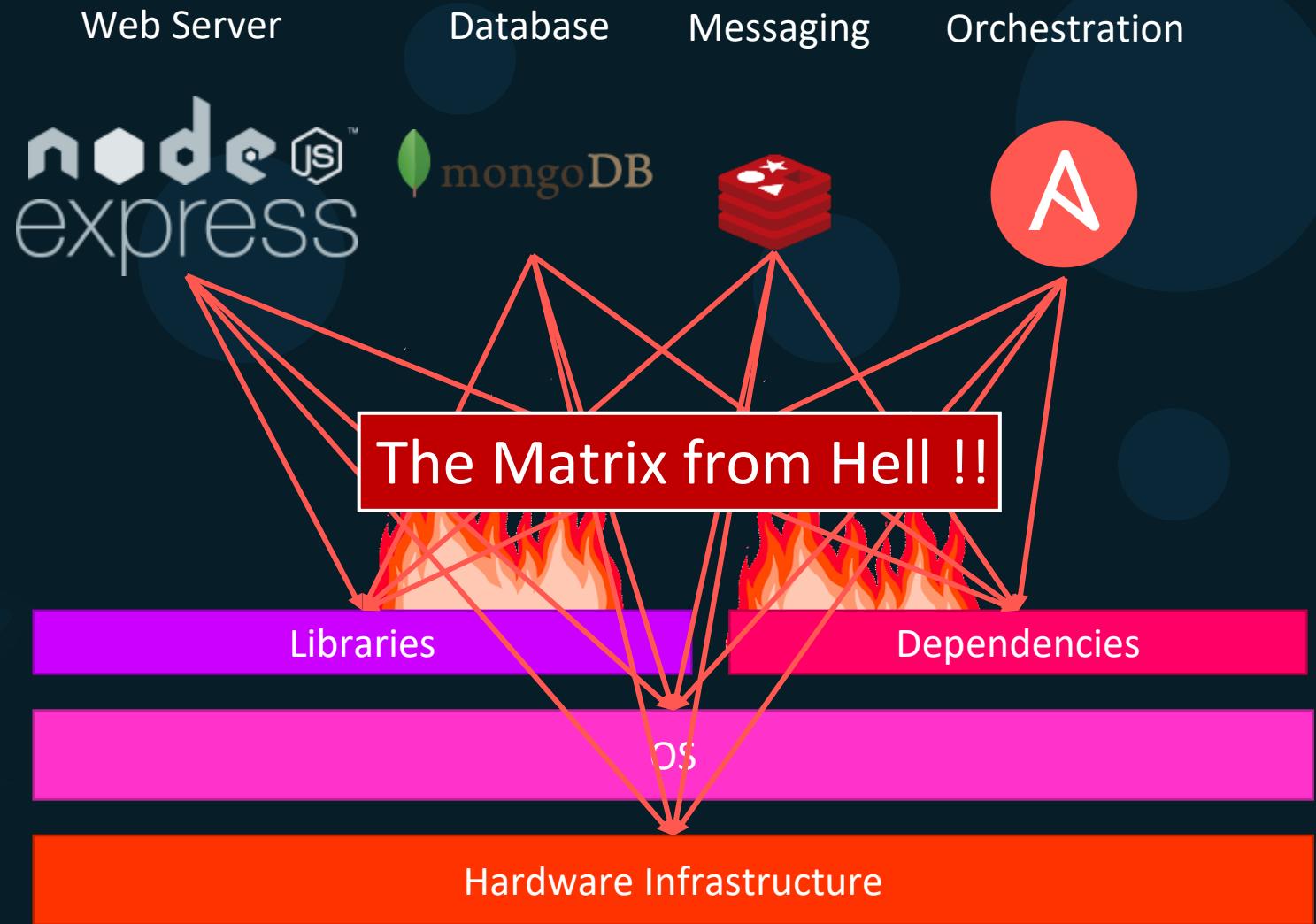
- Docker Swarm
- Docker vs Kubernetes

docker

overview

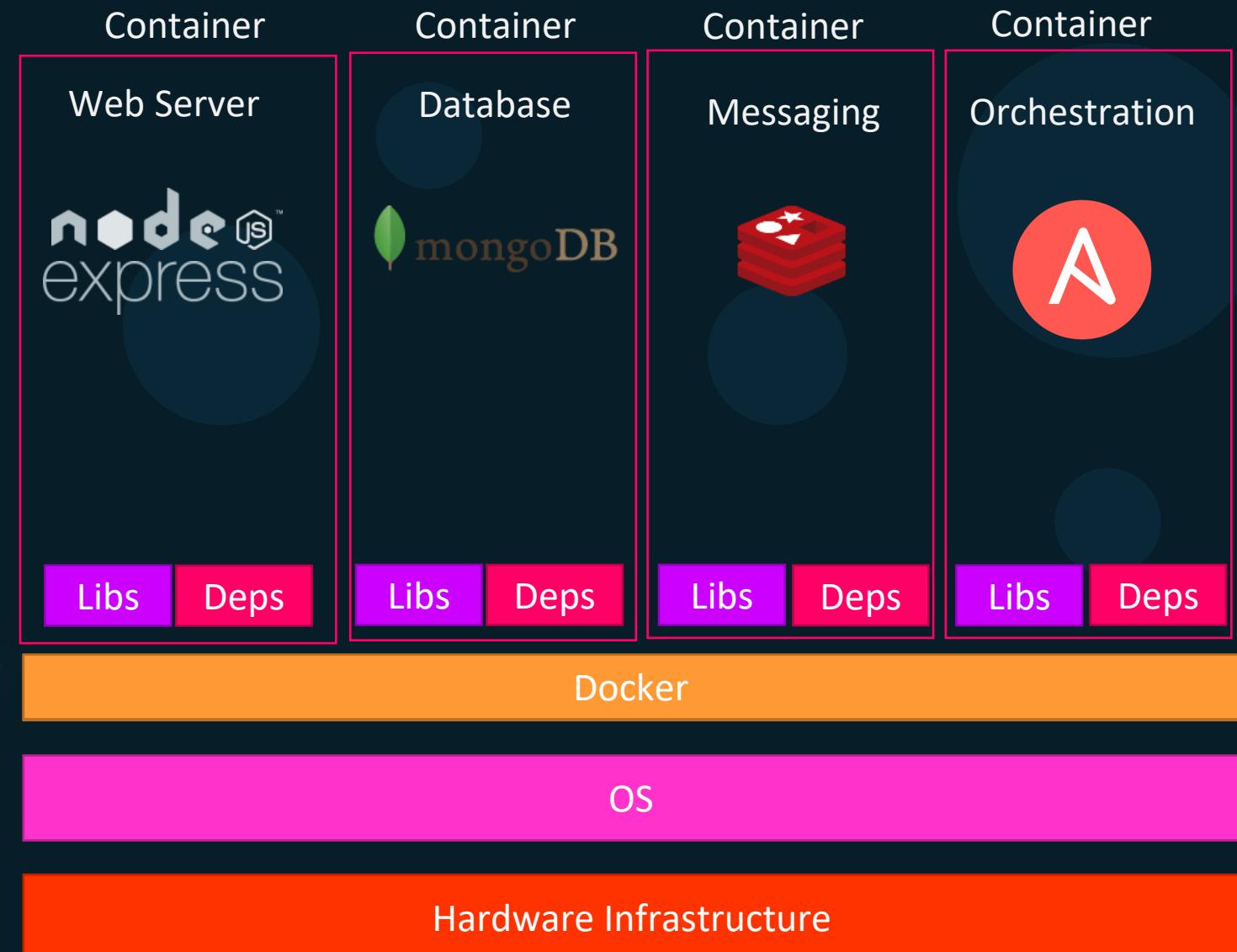
Why do you need docker?

- Compatibility/Dependency
- Long setup time
- Different Dev/Test/Prod environments

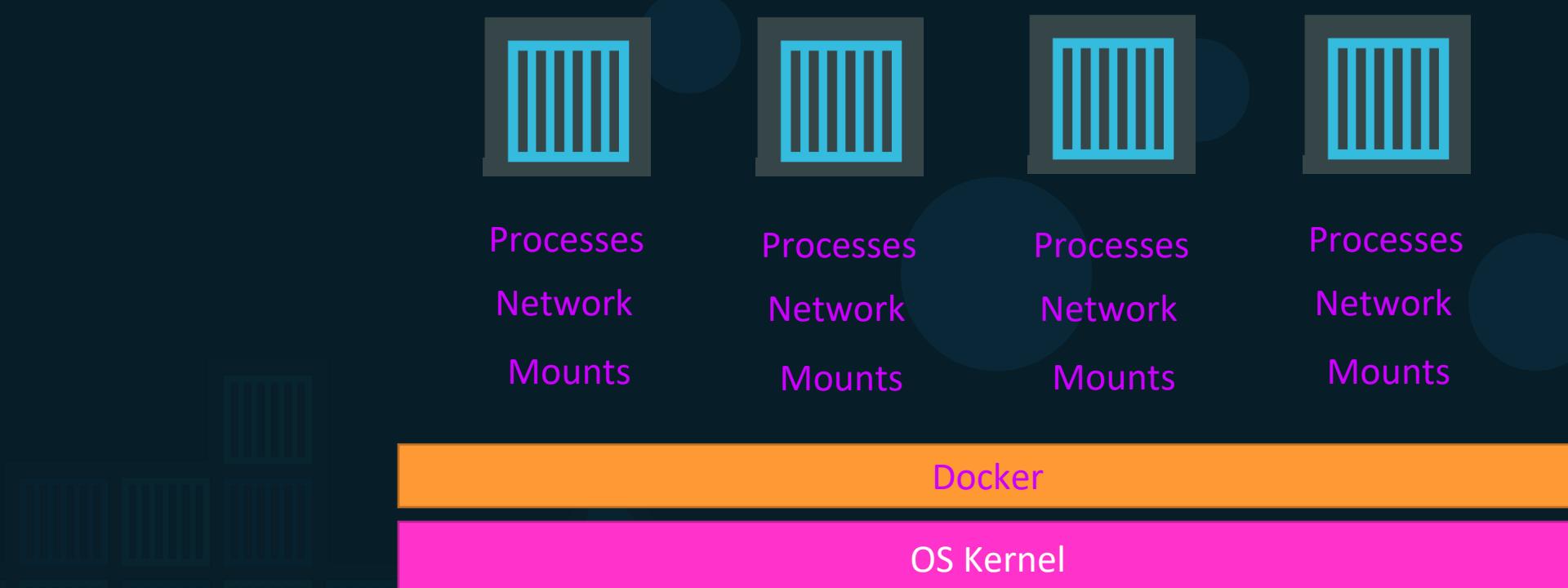


What can it do?

- Containerize Applications
- Run each service with its own dependencies in separate containers



What are containers?



Operating System



OS

Software

Software

Software

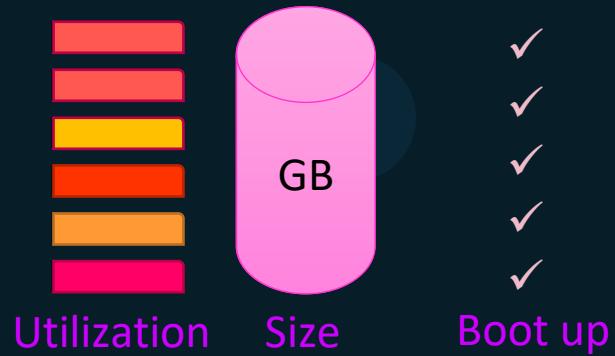
Software

OS Kernel

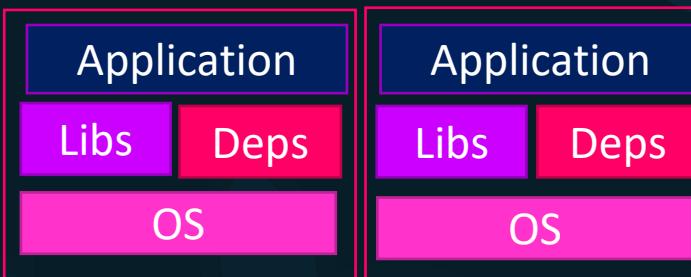
Sharing the kernel



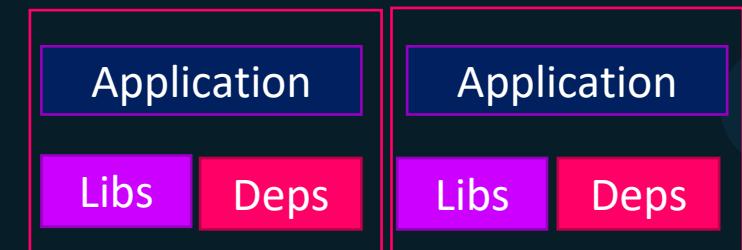
Containers vs Virtual Machines



Virtual Machine Virtual Machine



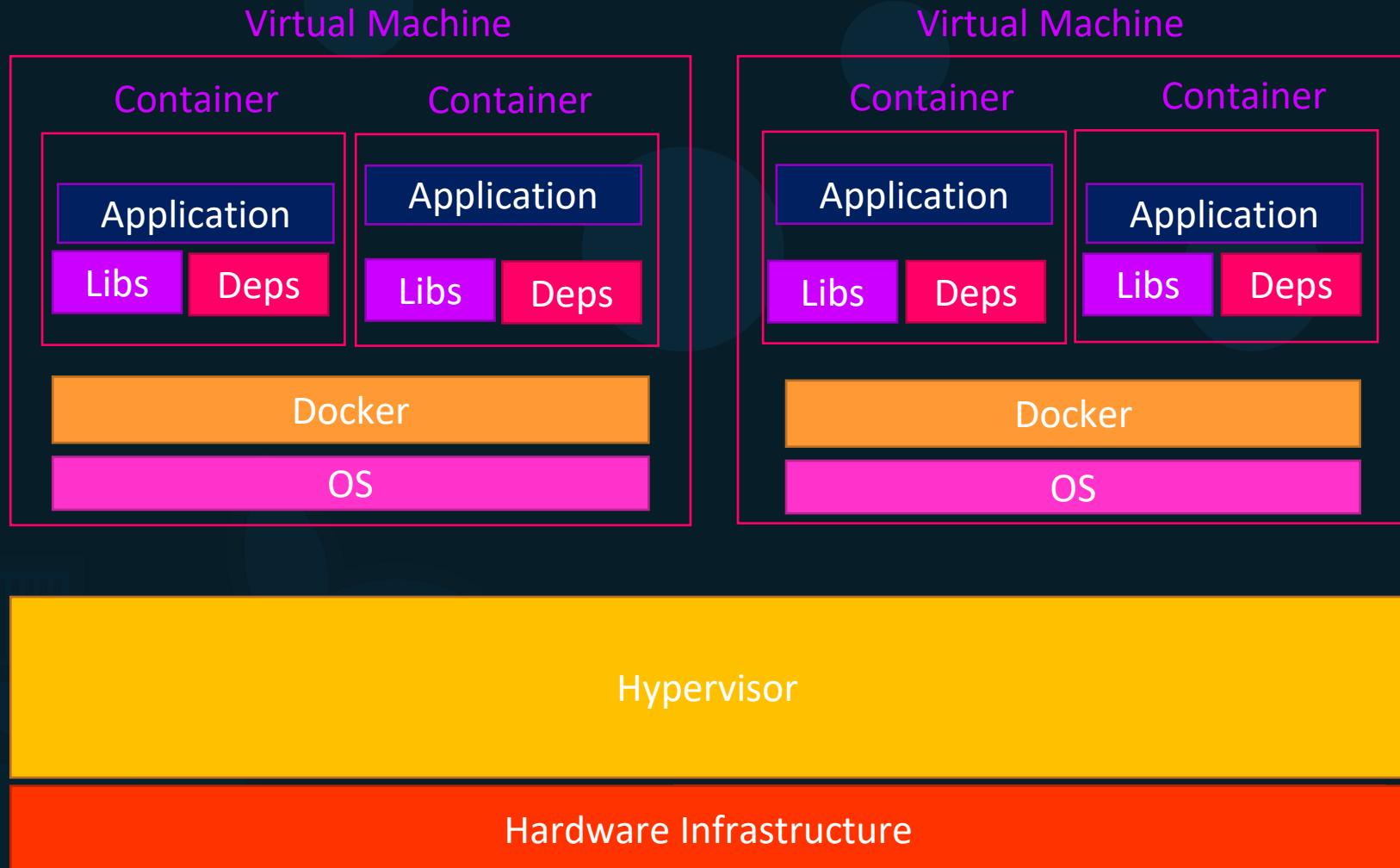
Container Container



Hardware Infrastructure

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pbUu>

Containers & Virtual Machines



How is it done?

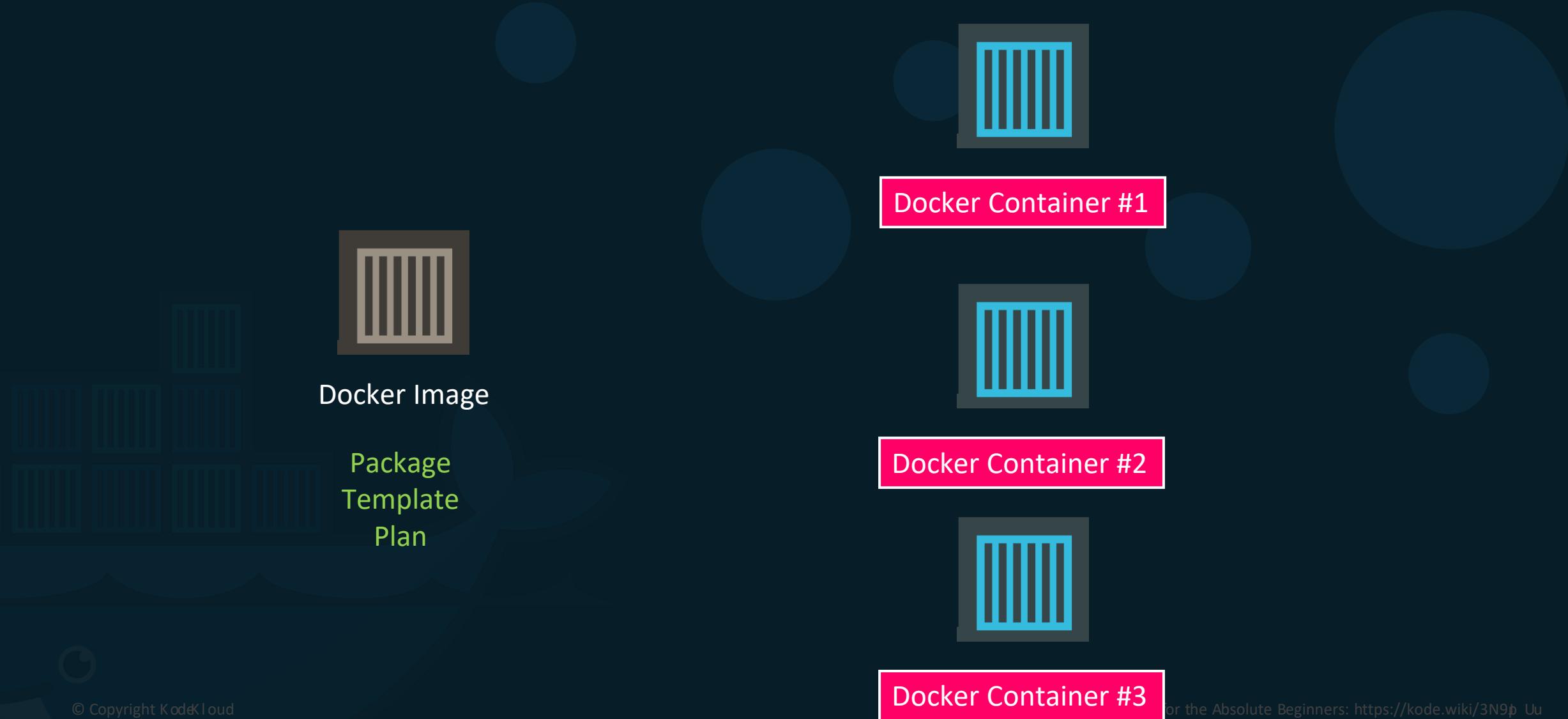
```
docker run ansible  
docker run mongodb  
docker run redis  
docker run nodejs  
docker run nodejs  
docker run nodejs
```



Public Docker registry - dockerhub



Container vs image



Docker in DevOps



Developer



DevOps



Operations



KodeKloud

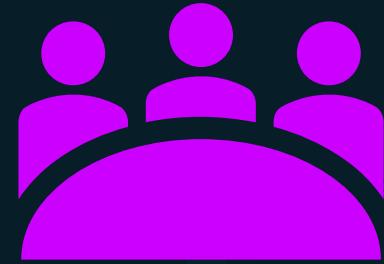
Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pJUu>



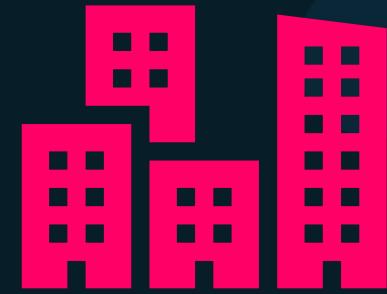
Getting Started

d o c k e r

Docker Editions



Community Edition



Enterprise Edition

Community Edition



Linux



MAC



Windows



Cloud

Community Edition



Linux



MAC



Windows



Cloud

Community Edition



Linux



MAC



Windows



Cloud



Install Docker

d e m o



KodeKloud

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pJUu>



docker

commands

Run – start a container

```
▶ docker run nginx
```

```
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
fc7181108d40: Already exists
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest:
sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
Status: Downloaded newer image for nginx:latest
```

ps – list containers

```
▶ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
796856ac413d	nginx	"nginx -g 'daemon of..."	7 seconds ago	Up 6 seconds	80/tcp	silly_sammet

```
▶ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	NAMES
796856ac413d	nginx	"nginx -g 'daemon of..."	7 seconds ago	Up 6 seconds	silly_sammet
cff8ac918a2f	redis	"docker-entrypoint.s..."	6 seconds ago	Exited (0) 3 seconds ago	relaxed_aryabhatta

STOP – stop a container

```
▶ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
796856ac413d	nginx	"nginx -g 'daemon of..."	7 seconds ago	Up 6 seconds	80/tcp	silly_sammet

```
▶ docker stop silly_sammet
```

```
silly_sammet
```

```
▶ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	NAMES
796856ac413d	nginx	"nginx -g 'daemon of..."	7 seconds ago	Exited (0) 3 seconds ago	silly_sammet
cff8ac918a2f	redis	"docker-entrypoint.s..."	6 seconds ago	Exited (0) 3 seconds ago	relaxed_aryabhata

Rm – Remove a container

```
▶ docker rm silly_sammet
```

```
silly_sammet
```

```
▶ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	NAMES
cff8ac918a2f	redis	"docker-entrypoint.s..."	6 seconds ago	Exited (0) 3 seconds ago	relaxed_aryabhata

images – List images

```
▶ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx	latest	f68d6e55e065	4 days ago	109MB
redis	latest	4760dc956b2d	15 months ago	107MB
ubuntu	latest	f975c5035748	16 months ago	112MB
alpine	latest	3fd9065eaf02	18 months ago	4.14MB

rmi – Remove images

```
▶ docker rmi nginx
```

```
Untagged: nginx:latest
Untagged: nginx@sha256:96fb261b66270b900ea5a2c17a26abbfabef5506e73c3a3c65869a6dbe83223a
Deleted: sha256:f68d6e55e06520f152403e6d96d0de5c9790a89b4cfcc99f4626f68146fa1dbdc
Deleted: sha256:1b0c768769e2bb66e74a205317ba531473781a78b77feef8ea6fd7be7f4044e1
Deleted: sha256:34138fb60020a180e512485fb96fd42e286fb0d86cf1fa2506b11ff6b945b03f
Deleted: sha256:cf5b3c6798f77b1f78bf4e297b27cf5b6caa982f04caeb5de7d13c255fd7a1e
```

! Delete all dependent containers to remove image

Pull – download an image

```
▶ docker run nginx
```

```
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
fc7181108d40: Already exists
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest:
sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
Status: Downloaded newer image for nginx:latest
```

```
▶ docker pull nginx
```

```
Using default tag: latest
latest: Pulling from library/nginx
fc7181108d40: Pull complete
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest:
sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
Status: Downloaded newer image for nginx:latest
```

```
▶ docker run ubuntu
```

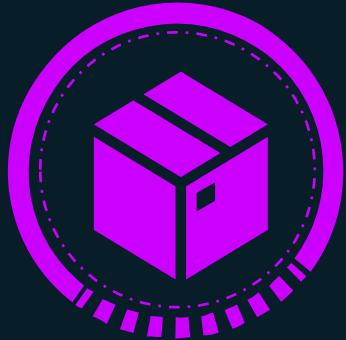
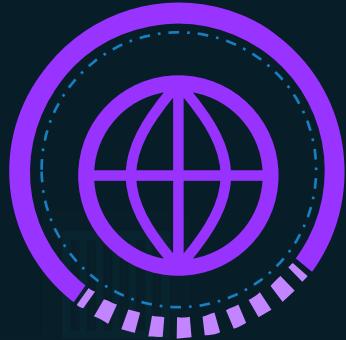
```
▶ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
--------------	-------	---------	---------	--------	-------

```
▶ docker ps -a
```

CONTAINER ID 45aacca36850	IMAGE ubuntu	COMMAND "/bin/bash"	CREATED 43 seconds ago	STATUS Exited (0) 41 seconds ago	PORTS
------------------------------	-----------------	------------------------	---------------------------	-------------------------------------	-------

```
▶ docker run ubuntu
```



```
▶ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
--------------	-------	---------	---------	--------	-------

```
▶ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
45aacca36850	ubuntu	"/bin/bash"	43 seconds ago	Exited (0) 41 seconds ago	

Append a command

```
▶ docker run ubuntu
```

```
▶ docker run ubuntu sleep 5
```



Exec – execute a command

```
▶ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	NAMES
538d037f94a7	ubuntu	"sleep 100"	6 seconds ago	Up 4 seconds	distracted_mcclintock

```
▶ docker exec distracted_mcclintock cat /etc/hosts
```

```
127.0.0.1      localhost
::1      localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.18.0.2      538d037f94a7
```

Run – attach and detach

```
▶ docker run kodekloud/simple-webapp
```

```
This is a sample web application that displays a colored background.  
* Serving Flask app "app" (lazy loading)  
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

```
▶ docker run -d kodekloud/simple-webapp
```

```
a043d40f85fe fa414254e4775f9336ea59e19e5cf597af5c554e0a35a1631118
```

```
▶ docker attach a043d
```



KodeKloud

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pJUu>



docker run

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pbUu>

Run – tag

```
docker run ubuntu
```

```
root@osboxes:/root # docker run ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
d5c6f90da05d: Pull complete
1300883d87d5: Pull complete
c220aa3cfclb: Pull complete
2e9398f099dc: Pull complete
dc27a084064f: Pull complete
Digest: sha256:34471448724419596ca4e890496d375801de21b0e67b81a77fd6155ce001edad
Status: Downloaded newer image for ubuntu:latest
root@osboxes:/root #
```

```
docker run Ubuntu:17.04
```

TAG

```
root@osboxes:/root # docker run ubuntu:17.04
Unable to find image 'ubuntu:17.04' locally
17.04: Pulling from library/ubuntu
e8a74323e913: Pull complete
5fe91835aea8: Pull complete
4b2aac3e93a5: Pull complete
faefbf4d7e6d: Pull complete
071e113a30b5: Pull complete
Digest: sha256:fe6cc21b9a65b07053cb2614f1ed8217a92cde64d0030d122e19d54881b6cb3e
Status: Downloaded newer image for ubuntu:17.04
```

Run – attach and detach

```
docker run mmumshad/simple-webapp
```

```
root@osboxes:/root/simple-webapp-docker # docker run mmumshad/simple-webapp
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
^C^C^C
```

```
docker run -d mmumshad/simple-webapp
```

```
root@osboxes:/root/simple-webapp-docker # docker run -d mmumshad/simple-webapp
754af299d82df776678cc511d76c790c2870525ca95d583fc9131f8eb85812f1
root@osboxes:/root/simple-webapp-docker #
```

```
docker attach sad_ramanujan
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
754af299d82d	mmumshad/simple-webapp	"/bin/sh -c 'FLASK..."	5 minutes ago	Up 5 minutes		sad_ramanujan

RUN - STDIN

```
root@osboxes:/root/simple-prompt-docker # ./app.sh  
Welcome! Please enter your name: Mumshad  
Hello and welcome Mumshad!
```

```
docker run mmumshad/simple-prompt-docker
```

```
root@osboxes:/root/simple-prompt-docker # docker run mmumshad/simple-prompt-docker  
Welcome! Please enter your name: Hello and welcome !
```

```
docker run -i mmumshad/simple-prompt-docker
```

```
root@osboxes:/root # docker run -i mmumshad/simple-prompt-docker  
Welcome! Please enter your name: Mumshad  
Hello and welcome Mumshad!
```

Run – PORT mapping

```
docker run mmumshad/simple-webapp
```

```
root@osboxes:/root/simple-webapp-docker # docker run mmumshad/simple-webapp
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
^C^C^C
```

http://172.17.0.2:5000

Internal IP

```
docker run -p 80:5000 mmumshad/simple-webapp
```

```
docker run -p 8000:5000 mmumshad/simple-webapp
```

```
docker run -p 8001:5000 mmumshad/simple-webapp
```

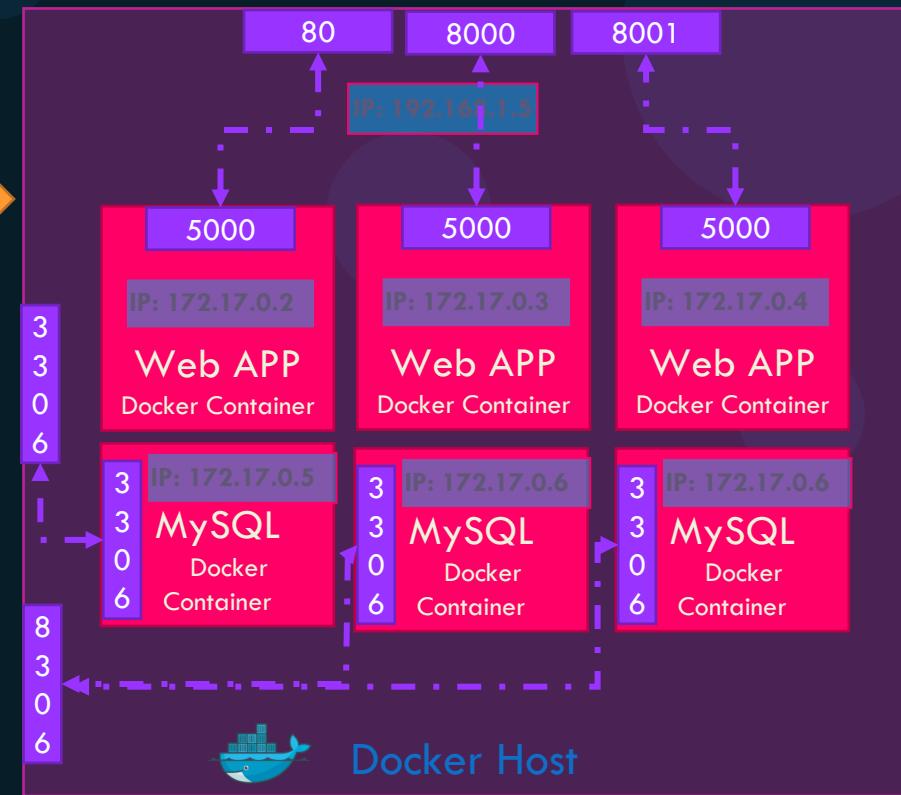
```
docker run -p 3306:3306 mysql
```

```
docker run -p 8306:3306 mysql
```

```
docker run -p 8306:3306 mysql
```



http://192.168.1.5:80



```
root@osboxes:/root # docker run -p 8306:3306 -e MYSQL_ROOT_PASSWORD=pass mysql
```

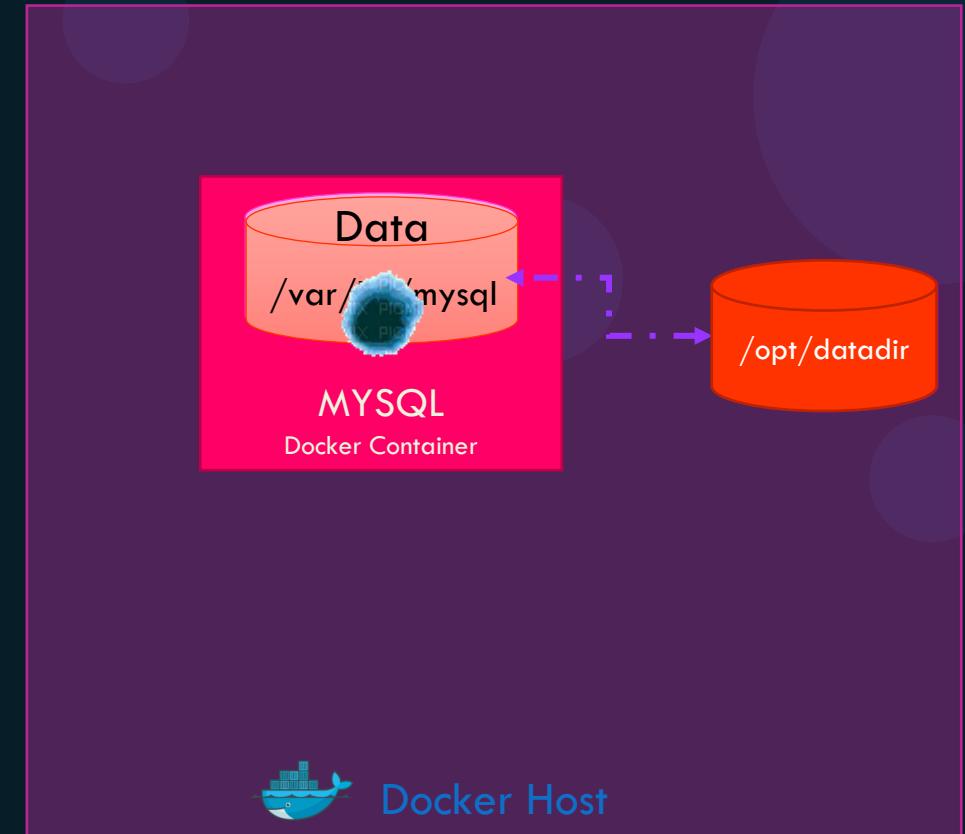
```
docker: Error response from daemon: driver failed programming external connectivity on endpoint boring_bhabha (5079d342b7e8ee11c71d46): Bind for 0.0.0.0:8306 failed: port is already allocated.
```

RUN – Volume mapping

```
docker run mysql
```

```
docker stop mysql  
docker rm mysql
```

```
docker run -v /opt/datadir:/var/lib/mysql mysql
```





KodeKloud

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pJUu>



d o c k e r i m a g e s

Check out our full course on Docker for the Absolute Beginners: https://kode.wiki/3N9p_Uu

What am I containerizing?



How to create my own image?

Dockerfile

```
FROM Ubuntu

RUN apt-get update
RUN apt-get install python

RUN pip install flask
RUN pip install flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

1. OS - Ubuntu

2. Update apt repo

3. Install dependencies using apt

4. Install Python dependencies using pip

5. Copy source code to /opt folder

6. Run the web server using “flask” command

```
docker build Dockerfile -t mmumshad/my-custom-app
```

```
docker push mmumshad/my-custom-app
```

Docker
Registry

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pbUu>



Dockerfile



```
Dockerfile
FROM Ubuntu
RUN apt-get update
RUN apt-get install python
RUN pip install flask
RUN pip install flask-mysql
COPY . /opt/source-code
ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

Start from a base OS or another image

Install all dependencies

Copy source code

Specify Entrypoint

Layered architecture

Dockerfile

```
FROM Ubuntu

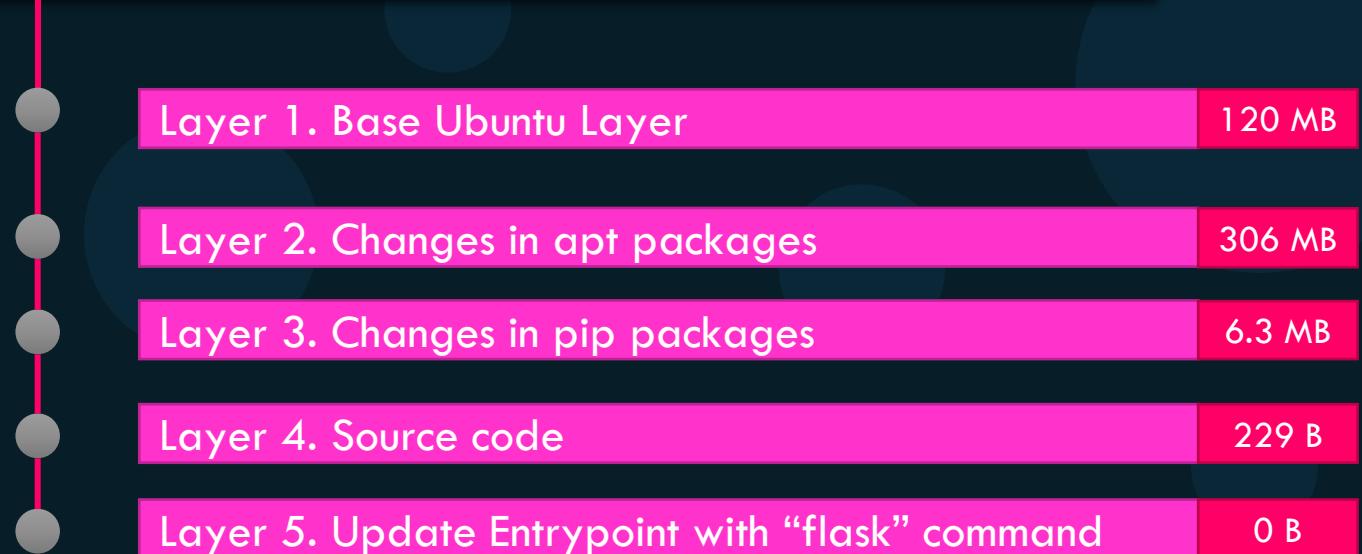
RUN apt-get update && apt-get -y install python

RUN pip install flask flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

```
docker build Dockerfile -t mmumshad/my-custom-app
```



```
root@osboxes:/root/simple-webapp-docker # docker history mmumshad/simple-webapp
IMAGE          CREATED      CREATED BY
1a45ba829f10  About an hour ago  /bin/sh -c #(nop) ENTRYPOINT ["/bin/sh" ... 0B
37d37ed8fe99  About an hour ago  /bin/sh -c #(nop) COPY file:29b92853d73898... 229B
d6aaebf8ded0  About an hour ago  /bin/sh -c pip install flask flask-mysql 6.39MB
e4c055538e60  About an hour ago  /bin/sh -c apt-get update && apt-get insta... 306MB
ccc7a11d65b1  2 weeks ago       /bin/sh -c #(nop) CMD ["/bin/bash"] 0B
<missing>      2 weeks ago       /bin/sh -c mkdir -p /run/systemd && echo '... 7B
<missing>      2 weeks ago       /bin/sh -c sed -i 's/^#\s*/(deb.*universe\... 2.76kB
<missing>      2 weeks ago       /bin/sh -c rm -rf /var/lib/apt/lists/* 0B
<missing>      2 weeks ago       /bin/sh -c set -xe  && echo '#!/bin/sh' >... 745B
<missing>      2 weeks ago       /bin/sh -c #(nop) ADD file:39d3593ea220e68... 120MB
```

Docker build output

```
root@osboxes:/root/simple-webapp-docker # docker build .
Sending build context to Docker daemon 3.072kB
Step 1/5 : FROM ubuntu
--> ccc7a11d65b1
Step 2/5 : RUN apt-get update && apt-get install -y python python-setuptools python-dev
--> Running in a7840dbfad17
Get:1 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Get:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security/universe Sources [46.3 kB]
Get:5 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Get:6 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [440 kB]
Step 3/5 : RUN pip install flask flask-mysql
--> Running in a4a6c9190ba3
Collecting flask
  Downloading Flask-0.12.2-py2.py3-none-any.whl (83kB)
Collecting flask-mysql
  Downloading Flask_SQLAlchemy-1.4.0-py2.py3-none-any.whl
Removing intermediate container a4a6c9190ba3
Step 4/5 : COPY app.py /opt/
--> e7cdab17e782
Removing intermediate container faaaaaf63c512
Step 5/5 : ENTRYPOINT FLASK_APP=/opt/app.py flask run --host=0.0.0.0
--> Running in d452c574a8bb
--> 9f27c36920bc
Removing intermediate container d452c574a8bb
Successfully built 9f27c36920bc
```

failure



Layer 1. Base Ubuntu Layer

Layer 2. Changes in apt packages

Layer 3. Changes in pip packages

Layer 4. Source code

Layer 5. Update Entrypoint with “flask” command

```
docker build Dockerfile -t mmumshad/my-custom-app
```

```
root@osboxes:/root/simple-webapp-docker # docker build .
Sending build context to Docker daemon 5.12kB
Step 1/5 : FROM ubuntu
--> ccc7a11d65b1
Step 2/5 : RUN apt-get update && apt-get install -y python python-pip
--> Using cache
--> e4c055538e60
Step 3/5 : RUN pip install flask
--> Running in aacdaccd7403
Collecting flask
  Downloading Flask-0.12.2-py2.py3-none-any.whl (83kB)
Removing intermediate container aacdaccd7403
Step 4/5 : COPY app.py /opt/
--> af41ef57f6f3
Removing intermediate container a49cc8befc8f
Step 5/5 : ENTRYPOINT FLASK_APP=/opt/app.py flask run --host=0.0.0.0
--> Running in 3d745ff07d5a
--> 910416d360b6
Removing intermediate container 3d745ff07d5a
Successfully built 910416d360b6
```

What can you containerize?



Containerize Everything!!!



KodeKloud

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pJUu>

docker environment variables

Environment Variables

app.py

```
import os
from flask import Flask

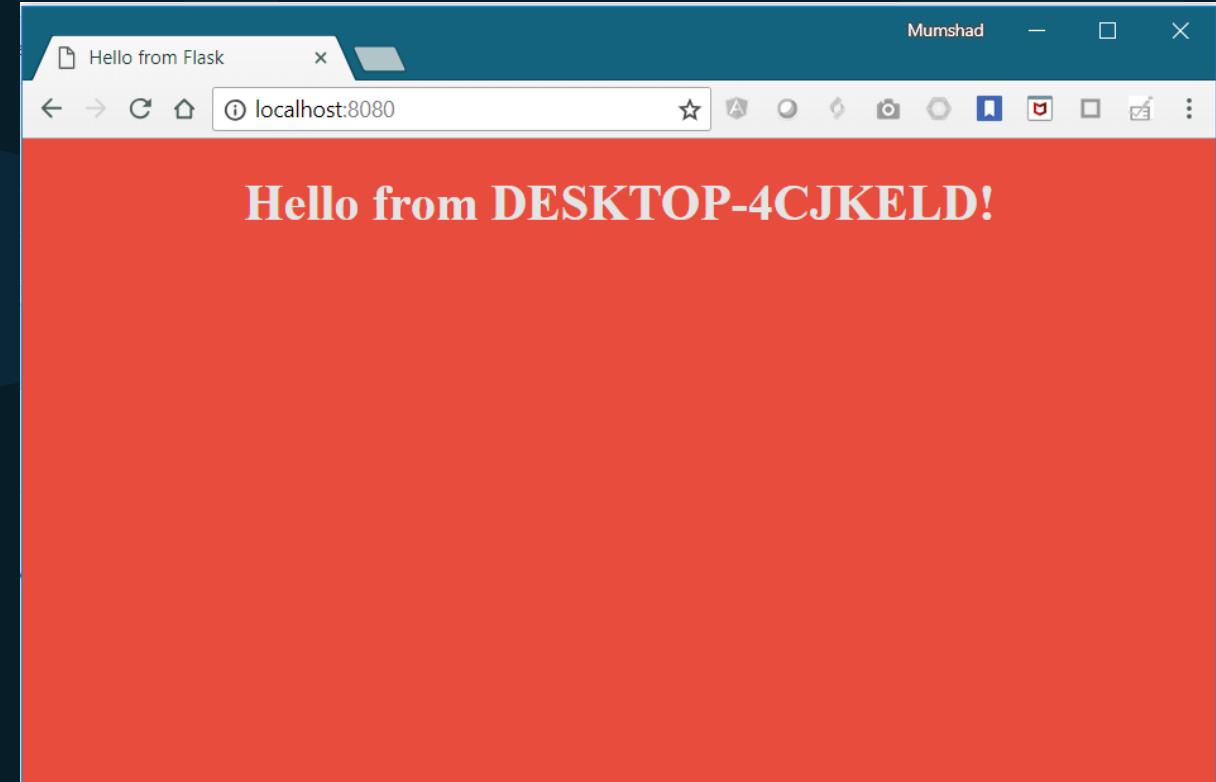
app = Flask(__name__)

...
...

color = "red"

@app.route("/")
def main():
    print(color)
    return render_template('hello.html', color=color)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port="8080")
```



▶ python app.py

Environment Variables

app.py

```
import os
from flask import Flask

app = Flask(__name__)

...
...

color = "red"

@app.route("/")
def main():
    print(color)
    return render_template('hello.html', color=color)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port="8080")
```

Environment Variables

app.py

```
import os
from flask import Flask

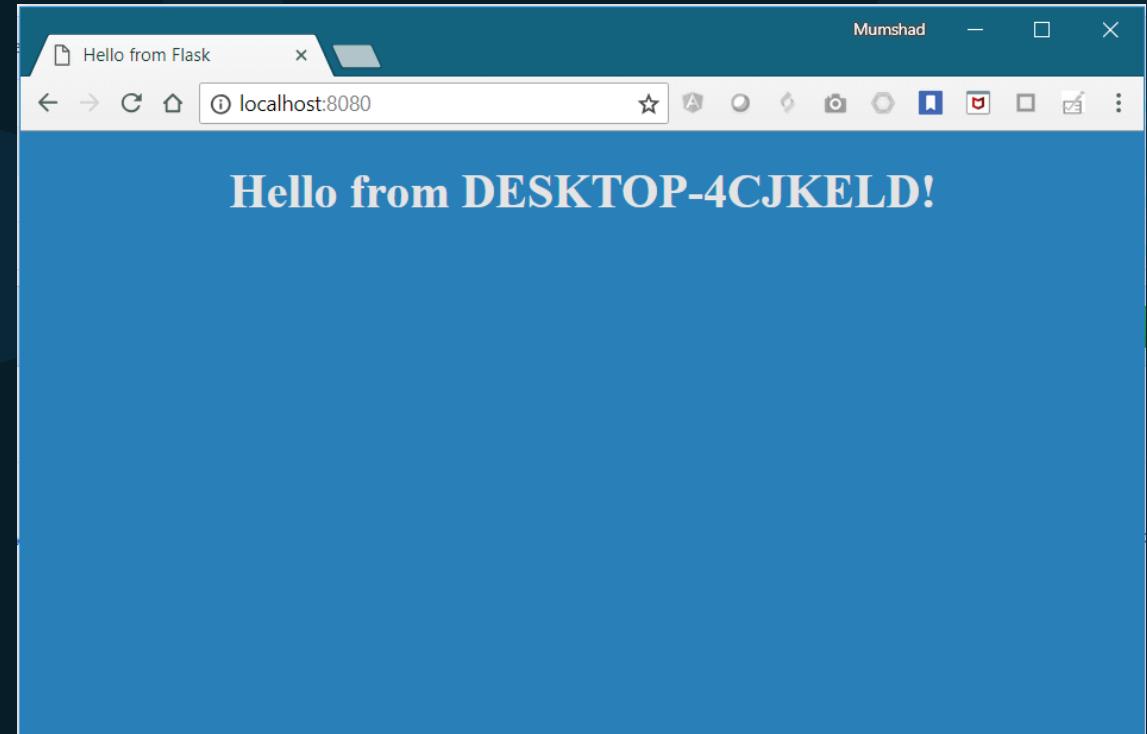
app = Flask(__name__)

...
...

color = os.environ.get('APP_COLOR')

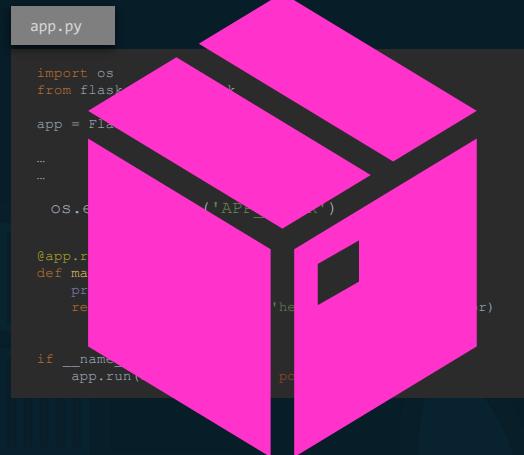
@app.route("/")
def main():
    print(color)
    return render_template('hello.html', color=color)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port="8080")
```

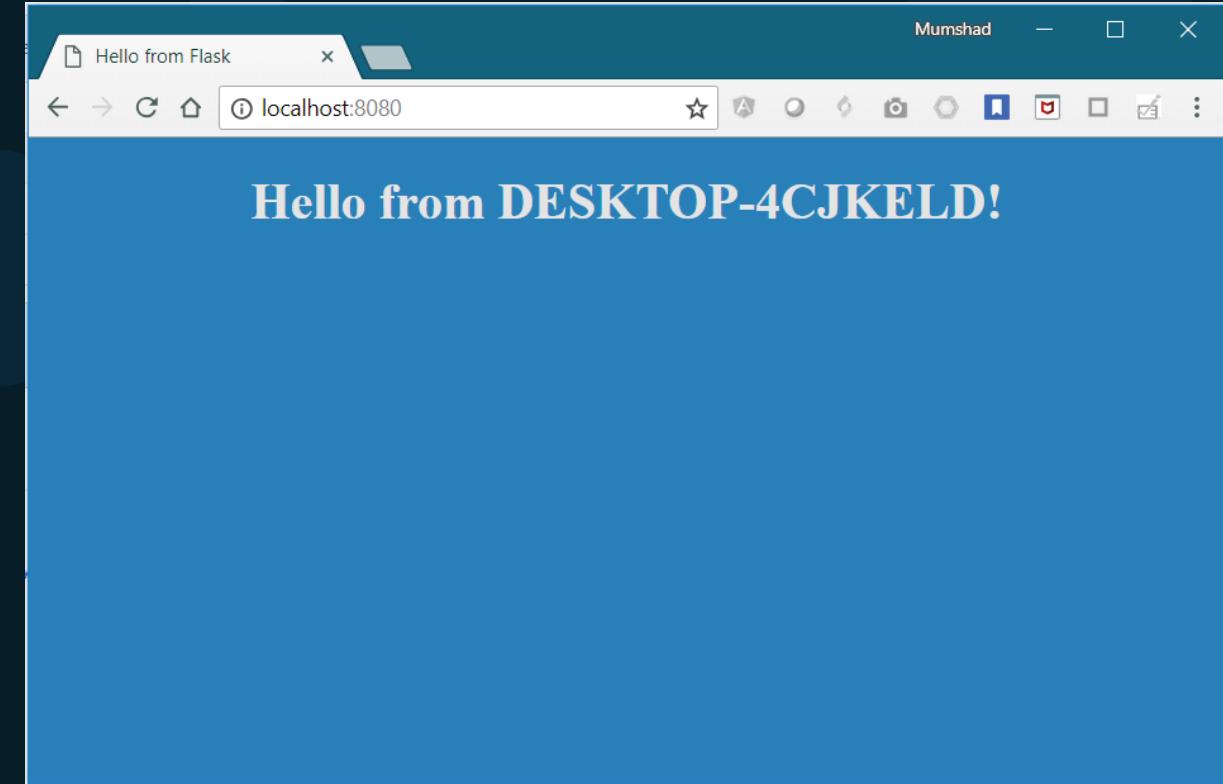


▶ export APP_COLOR=blue; python app.py

ENV Variables in Docker



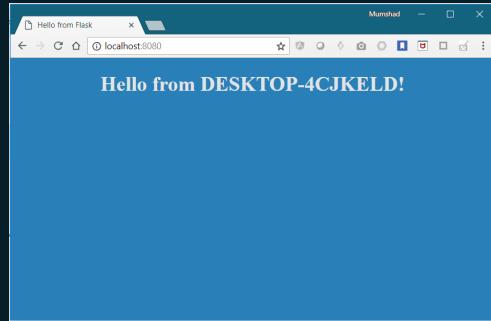
```
app.py
import os
from flask import Flask
app = Flask(__name__)
...
os.environ['APP_COLOR'] = 'blue'
@app.route('/')
def main():
    print('Hello from DESKTOP-4CJKELD!')
    return 'Hello from DESKTOP-4CJKELD!'
if __name__ == '__main__':
    app.run()
```



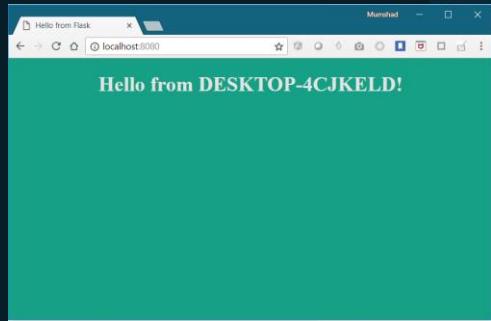
► docker run -e APP_COLOR=blue

ENV Variables in Docker

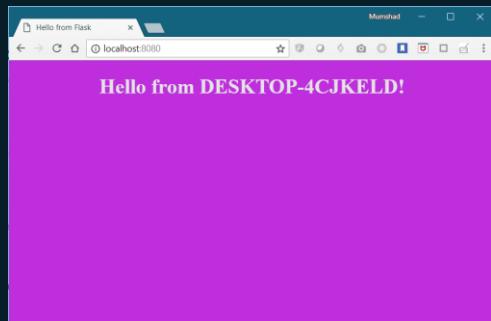
```
▶ docker run -e APP_COLOR=blue simple-webapp-color
```



```
▶ docker run -e APP_COLOR=green simple-webapp-color
```



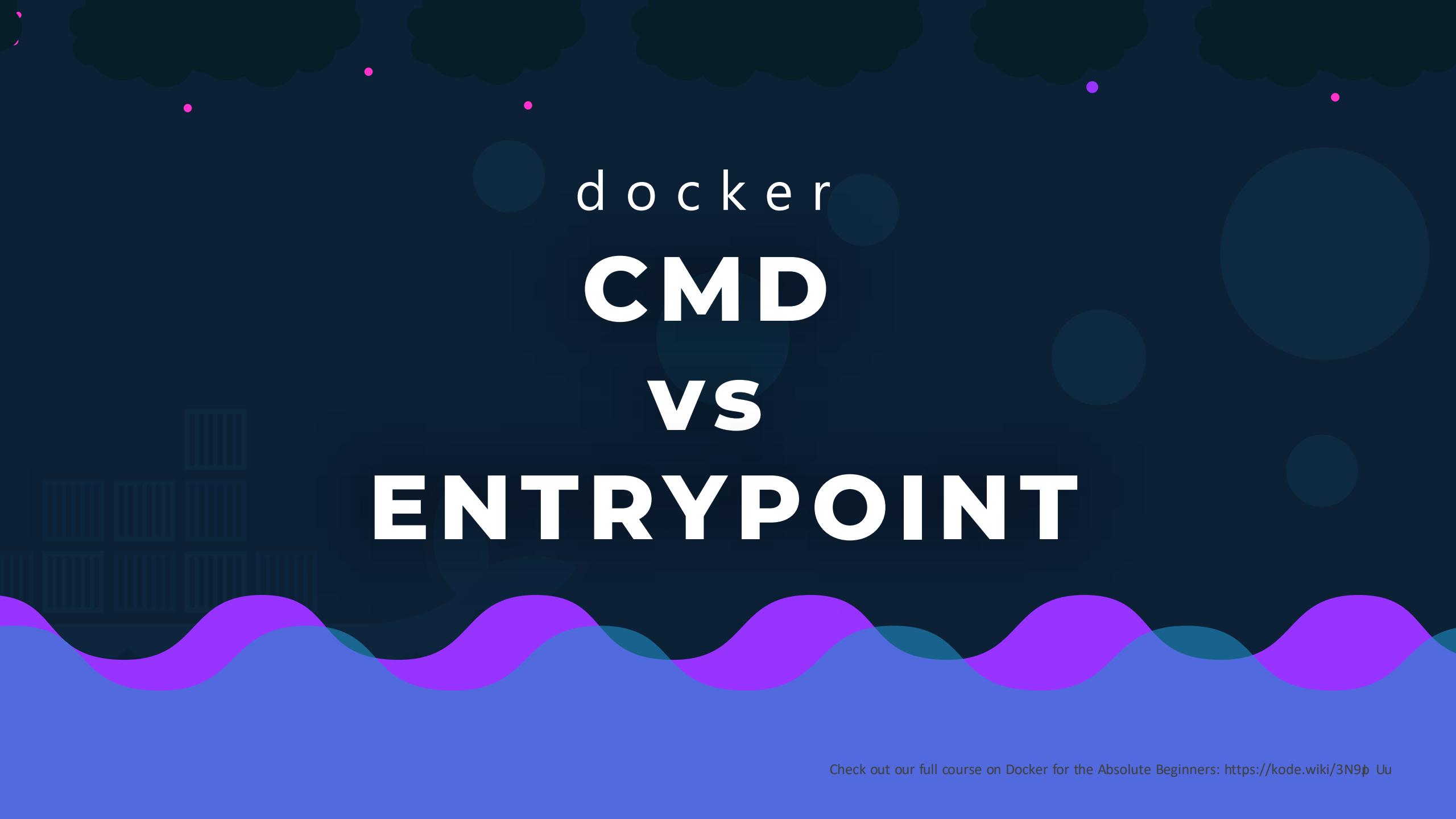
```
▶ docker run -e APP_COLOR=pink simple-webapp-color
```





KodeKloud

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pJUu>



docker **CMD** vs **ENTRYPOINT**

Check out our full course on Docker for the Absolute Beginners: https://kode.wiki/3N9p_Uu

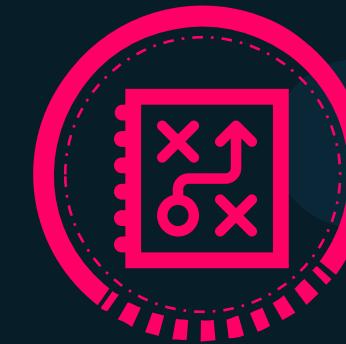
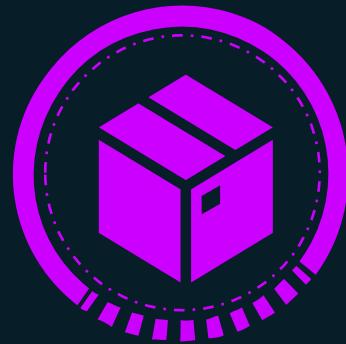
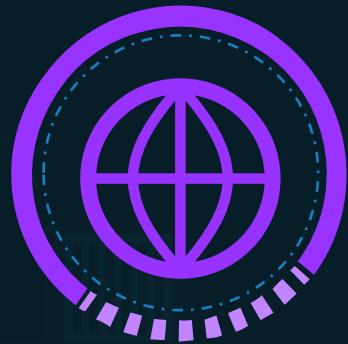
```
▶ docker run ubuntu
```

```
▶ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
--------------	-------	---------	---------	--------	-------

```
▶ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
45aacca36850	ubuntu	"/bin/bash"	43 seconds ago	Exited (0) 41 seconds ago	



```
# Install Nginx.  
RUN \  
    add-apt-repository -y ppa:nginx/stable && \  
    apt-get update && \  
    apt-get install -y nginx && \  
    rm -rf /var/lib/apt/lists/* && \  
    echo "\ndaemon off;" >> /etc/nginx/nginx.conf && \  
    chown -R www-data:www-data /var/lib/nginx  
  
# Define mountable directories.  
VOLUME ["/etc/nginx/sites-enabled", "/etc/nginx/certs", "/etc/nginx/cor  
  
# Define working directory.  
WORKDIR /etc/nginx  
  
# Define default command.  
CMD ["nginx"]  
  
ARG MYSQL_SERVER_PACKAGE_URL=https://repo.mysql.com/yum/mysql-8.0-community/docker/x86_64/  
ARG MYSQL_SHELL_PACKAGE_URL=https://repo.mysql.com/yum/mysql-tools-community/el/7/x86_64/  
  
# Install server  
RUN rpmkeys --import https://repo.mysql.com/RPM-GPG-KEY-mysql \  
    && yum install -y $MYSQL_SERVER_PACKAGE_URL $MYSQL_SHELL_PACKAGE_URL libpwquality \  
    && yum clean all \  
    && mkdir /docker-entrypoint-initdb.d  
  
VOLUME /var/lib/mysql  
  
COPY docker-entrypoint.sh /entrypoint.sh  
COPY healthcheck.sh /healthcheck.sh  
ENTRYPOINT ["/entrypoint.sh"]  
HEALTHCHECK CMD /healthcheck.sh  
EXPOSE 3306 33060  
CMD ["mysqld"]
```

```
# Pull base image.  
FROM ubuntu:14.04  
  
# Install.  
RUN \  
    sed -i 's/# \(.*multiverse$\)/\1/g' /etc/apt/sources.list && \  
    apt-get update && \  
    apt-get -y upgrade && \  
    apt-get install -y build-essential && \  
    apt-get install -y software-properties-common && \  
    apt-get install -y byobu curl git htop man unzip vim wget && \  
    rm -rf /var/lib/apt/lists/*  
  
# Add files.  
ADD root/.bashrc /root/.bashrc  
ADD root/.gitconfig /root/.gitconfig  
ADD root/.scripts /root/.scripts  
  
# Set environment variables.  
ENV HOME /root  
  
# Define working directory.  
WORKDIR /root  
  
# Define default command.  
CMD ["bash"]
```



▶ docker run ubuntu [COMMAND]

▶ docker run ubuntu sleep 5



1

FROM Ubuntu

CMD sleep 5

CMD command param1

CMD ["command", "param1"]

CMD sleep 5

CMD ["sleep", "5"]

CMD ["sleep 5"]



► docker build -t ubuntu-sleeper .

► docker run ubuntu-sleeper



4

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pbUu>

```
FROM Ubuntu
```

```
CMD sleep 5
```

Command at Startup: sleep 10

```
▶ docker run ubuntu-sleeper sleep 10
```

```
FROM Ubuntu
```

```
ENTRYPOINT sleep  
["sleep"]
```

Command at Startup:

```
▶ docker run ubuntu-sleeper
```

```
sleep: missing operand  
Try 'sleep --help' for more information.
```

Command at Startup:

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pbUu>

```
FROM Ubuntu
```

```
ENTRYPOINT ["sleep"]
```

```
CMD ["5"]
```

Command at Startup:

```
▶ docker run ubuntu-sleeper 10.0
```

Command at Startup:

```
▶ docker run --entrypoint sleep 2.0 ubuntu-sleeper 10
```

Command at Startup:

Please refer to the article on Docker for the Absolute Beginners: <https://kode.wiki/3N9pbUu>



KodeKloud

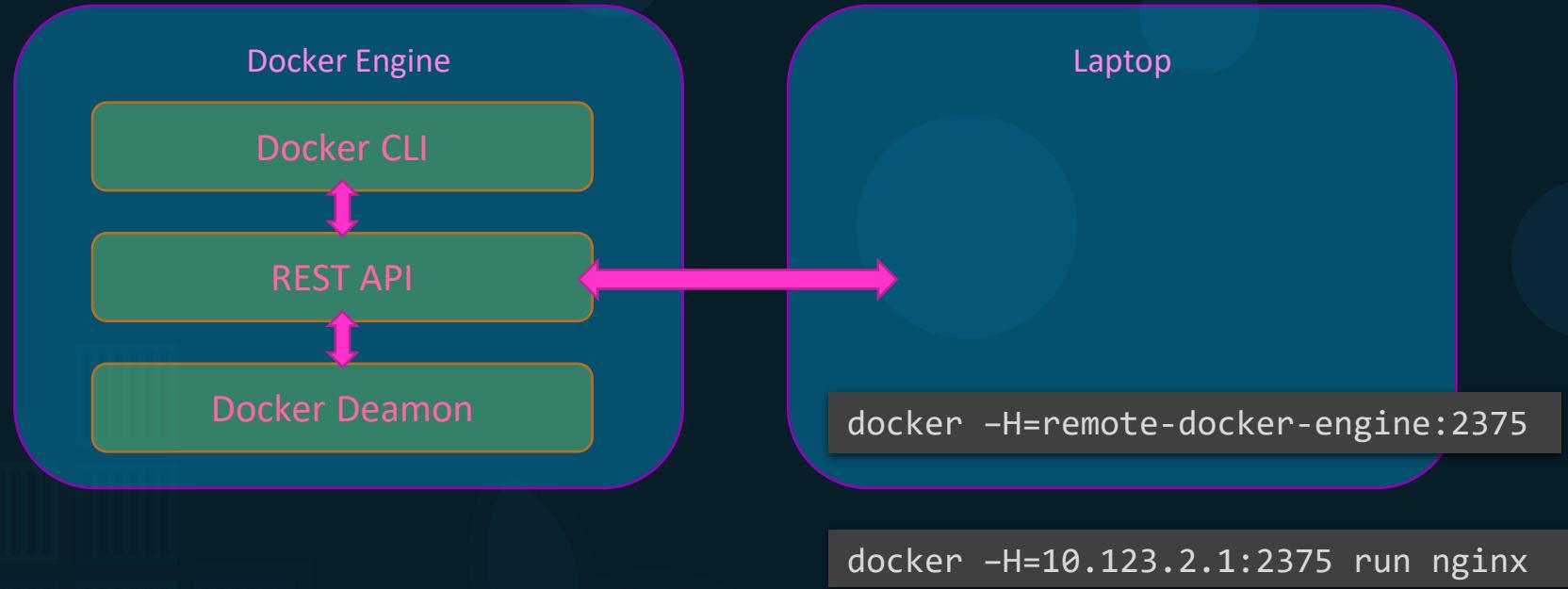
Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pJUu>



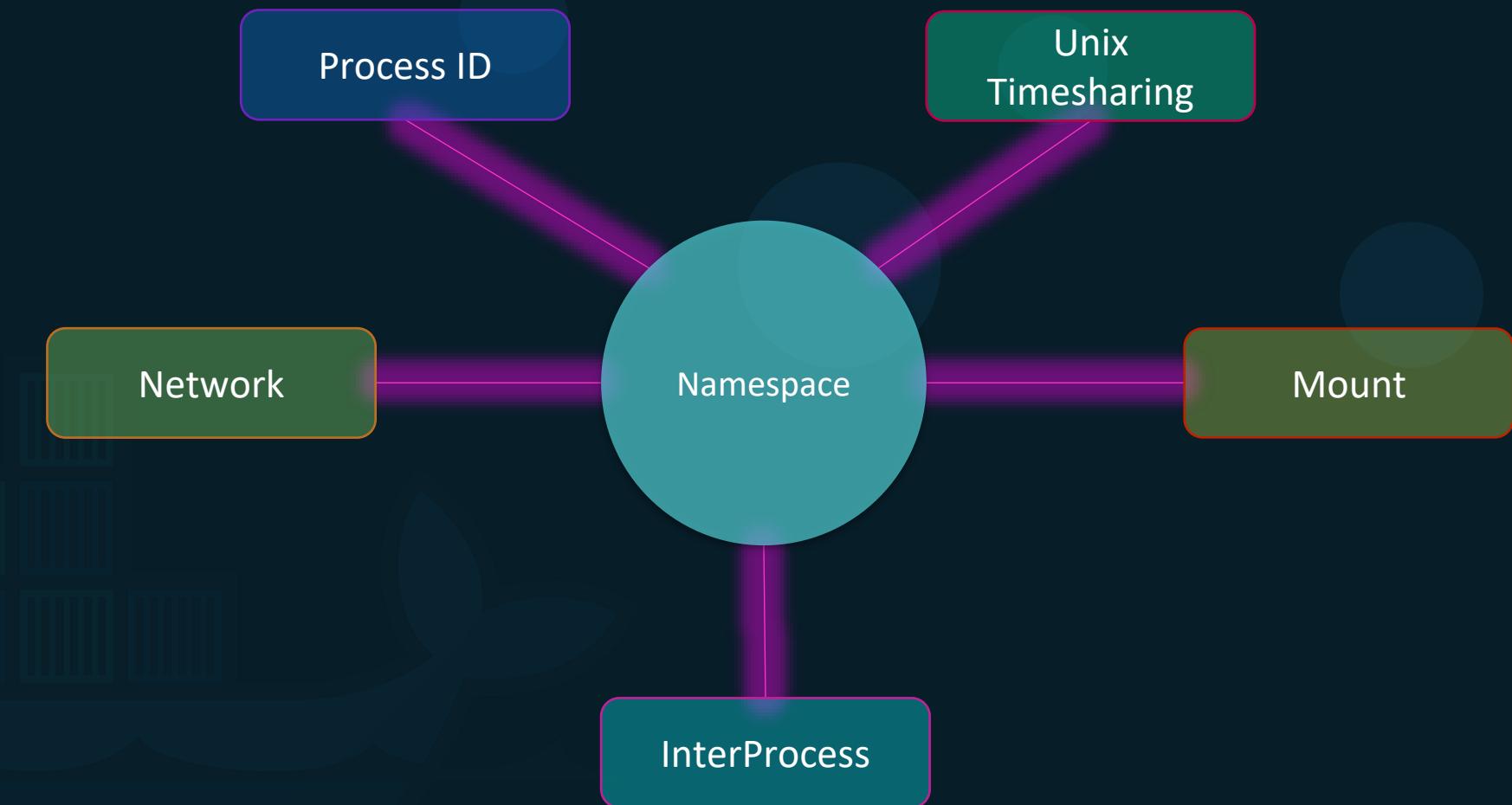
d o c k e r engine

Check out our full course on Docker for the Absolute Beginners: https://kode.wiki/3N9p_Uu

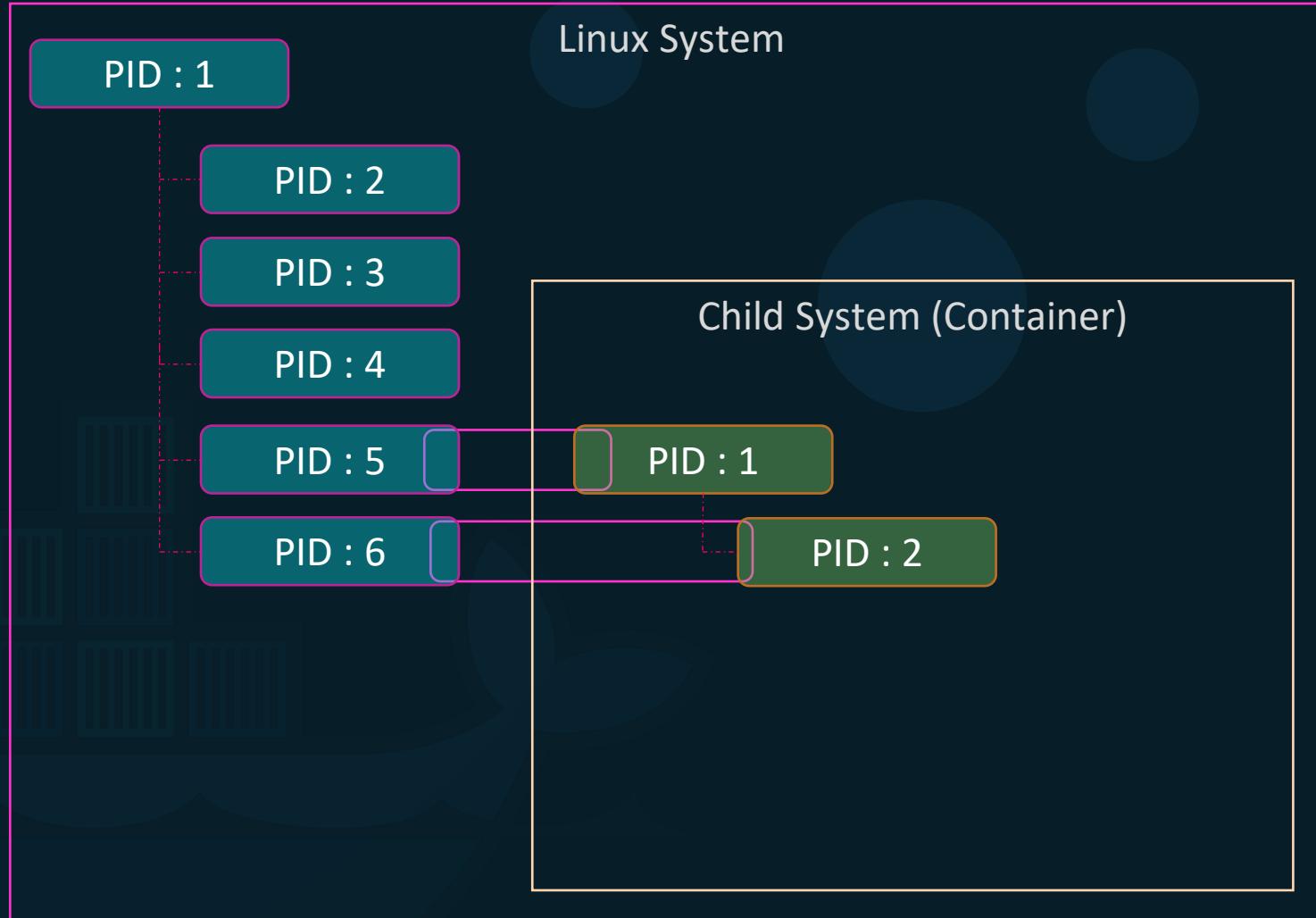
Docker Engine



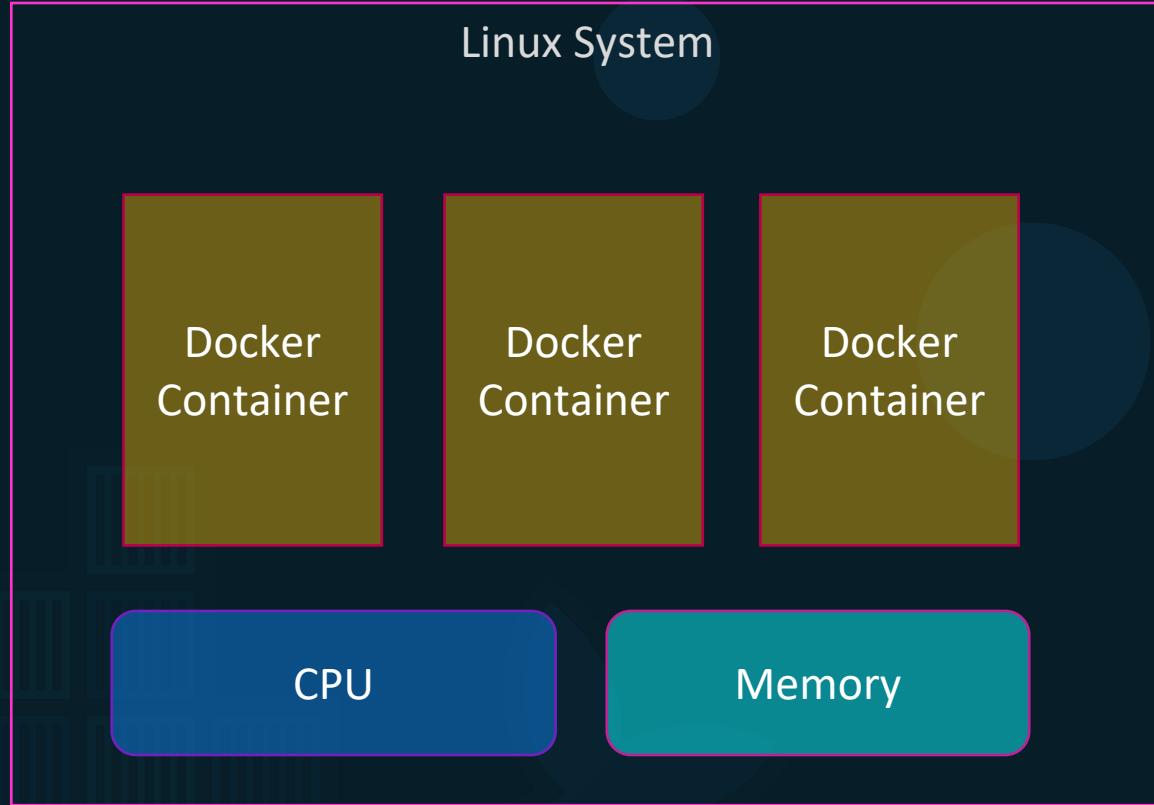
containerization



Namespace - PID



cgroups



```
docker run --cpus=.5 ubuntu
```

```
docker run --memory=100m ubuntu
```



KodeKloud

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pJUu>



d o c k e r storage

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pbUu>

File system

```
/var/lib/docker
  ├── aufs
  ├── containers
  ├── image
  └── volumes
```

Layered architecture

Dockerfile

```
FROM Ubuntu

RUN apt-get update && apt-get -y install python

RUN pip install flask flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

```
docker build Dockerfile -t mmumshad/my-custom-app
```

Dockerfile2

```
FROM Ubuntu

RUN apt-get update && apt-get -y install python

RUN pip install flask flask-mysql

COPY app2.py /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app2.py flask run
```

```
docker build Dockerfile2 -t mmumshad/my-custom-app-2
```

Layer 1. Base Ubuntu Layer 120 MB

Layer 2. Changes in apt packages 306 MB

Layer 3. Changes in pip packages 6.3 MB

Layer 4. Source code 229 B

Layer 5. Update Entrypoint 0 B

Layer 1. Base Ubuntu Layer 0 MB

Layer 2. Changes in apt packages 0 MB

Layer 3. Changes in pip packages 0 MB

Layer 4. Source code 229 B

Layer 5. Update Entrypoint 0 B

Layered architecture

Container Layer

Read Write

Layer 6. Container Layer

```
docker run mmumshad/my-custom-app
```

Image Layers

Read Only

Layer 5. Update Entrypoint with “flask” command

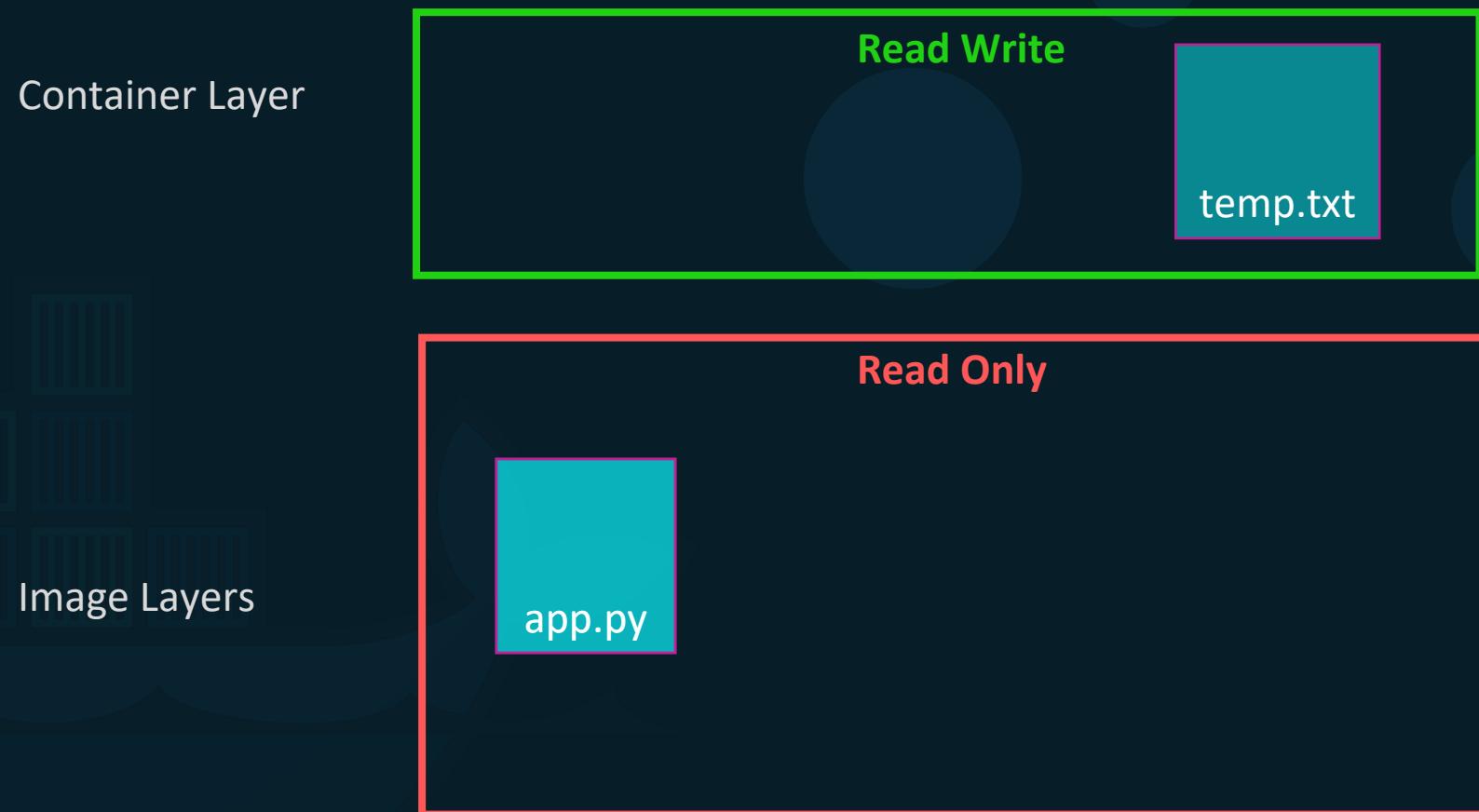
Layer 4. Source code

Layer 3. Changes in pip packages

Layer 2. Changes in apt packages

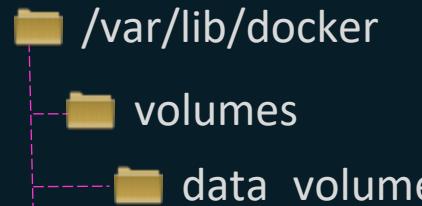
Layer 1. Base Ubuntu Layer

COPY-ON-WRITE



volumes

```
docker volume create data_volume
```

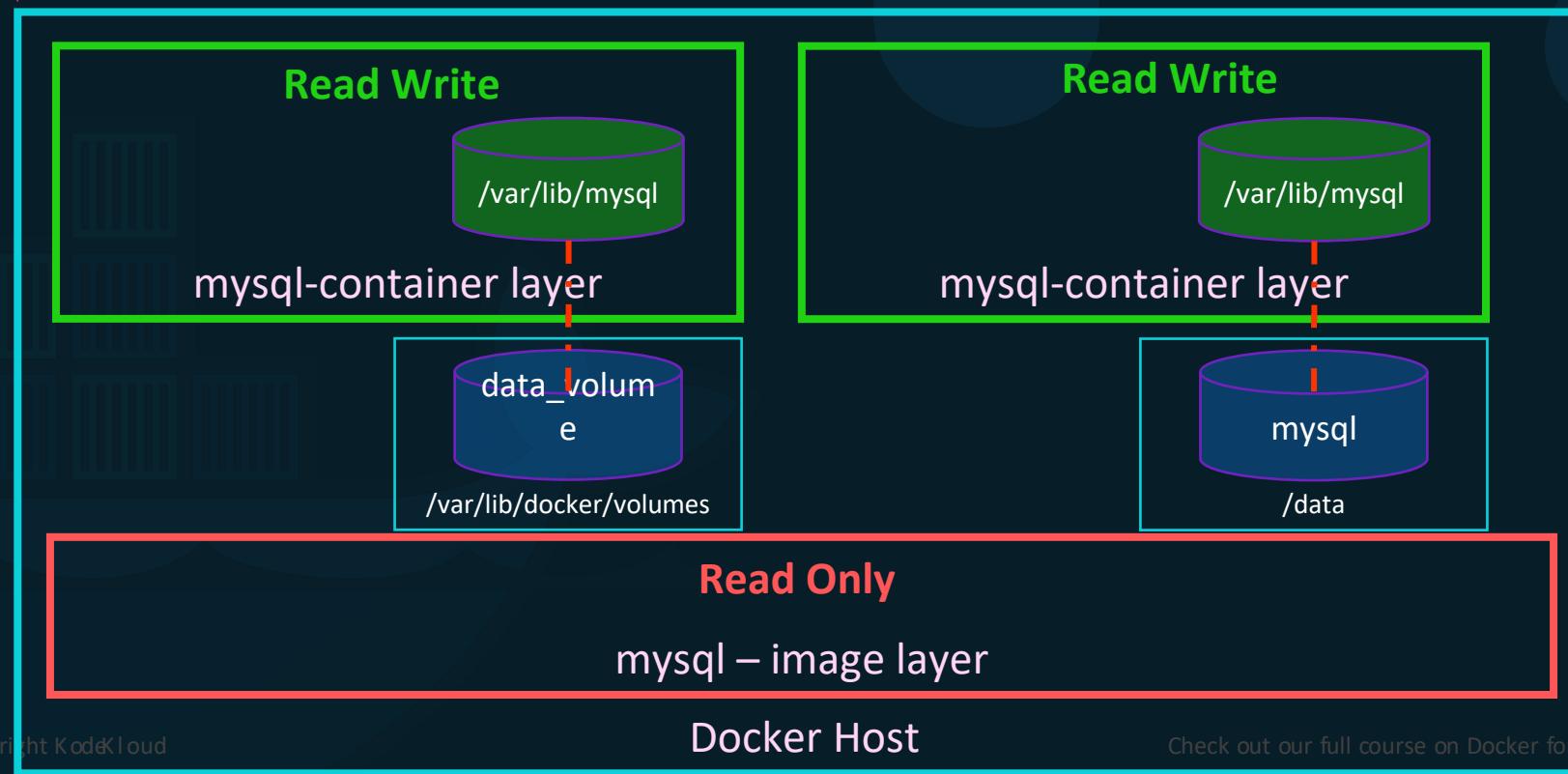


```
docker run -v data_volume:/var/lib/mysql mysql
```

```
docker run -v data_volume2:/var/lib/mysql mysql
```

```
docker run -v /data/mysql:/var/lib/mysql mysql
```

```
docker run \
--mount type=bind,source=/data/mysql,target=/var/lib/mysql mysql
```



Storage drivers

- AUFS
- ZFS
- BTRFS
- Device Mapper
- Overlay
- Overlay2



KodeKloud

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pJUu>



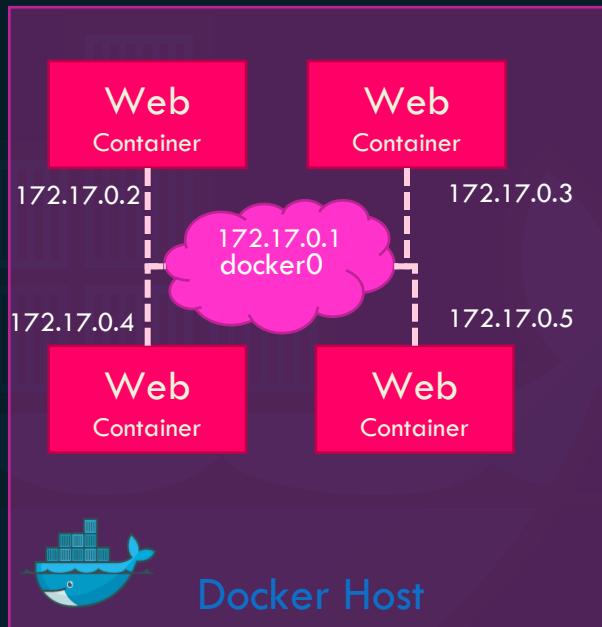
d o c k e r networking

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pUu>

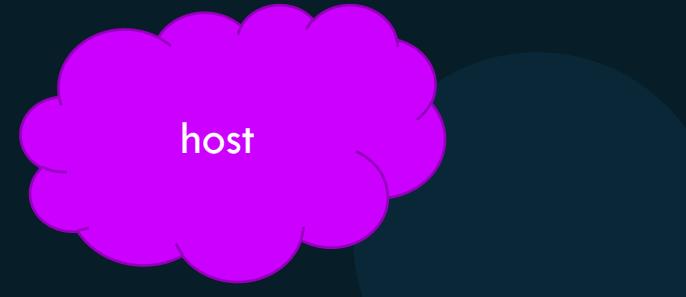
Default networks



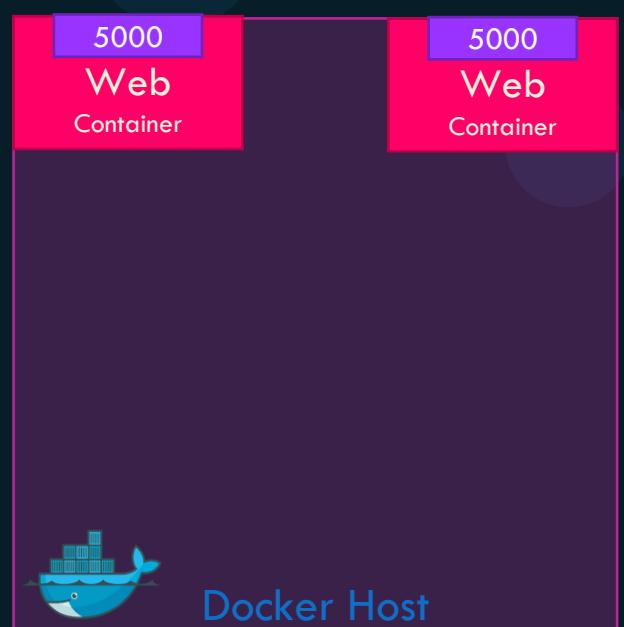
```
docker run ubuntu
```



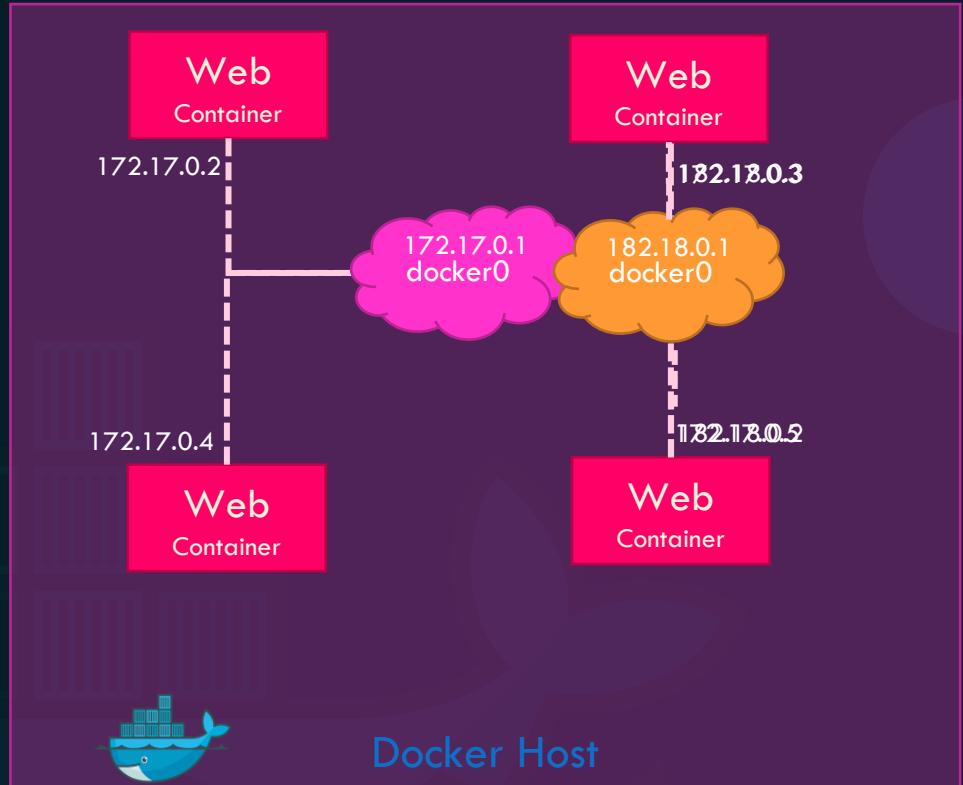
```
docker run Ubuntu --network=none
```



```
docker run Ubuntu --network=host
```



User-defined networks

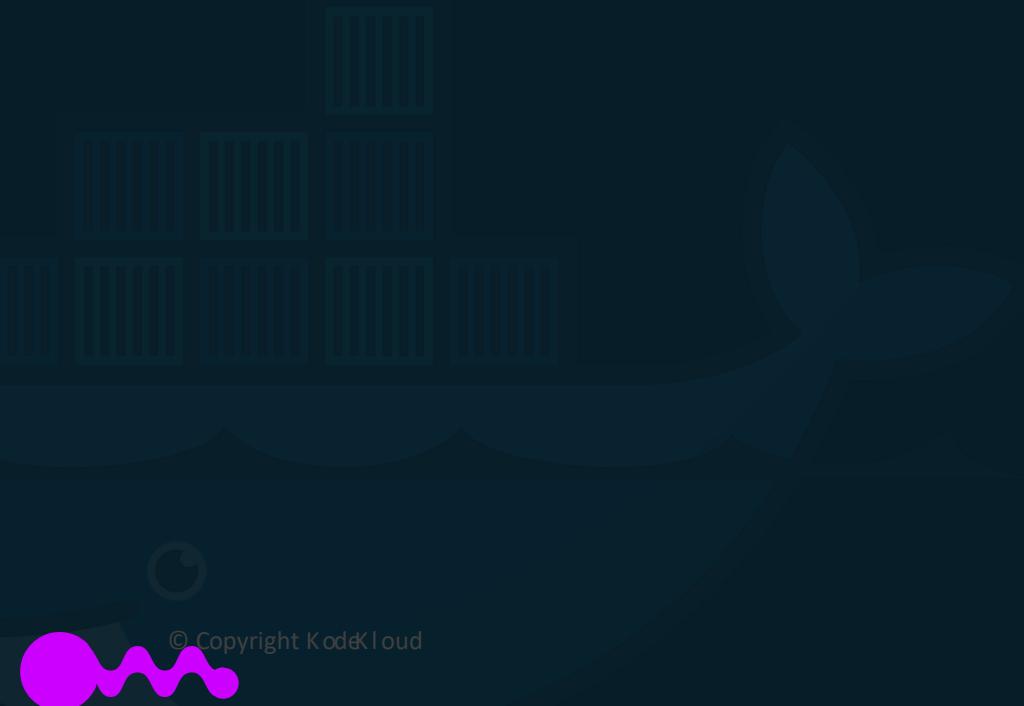


```
docker network create \
--driver bridge \
--subnet 182.18.0.0/16
custom-isolated-network
```

```
docker network ls
```

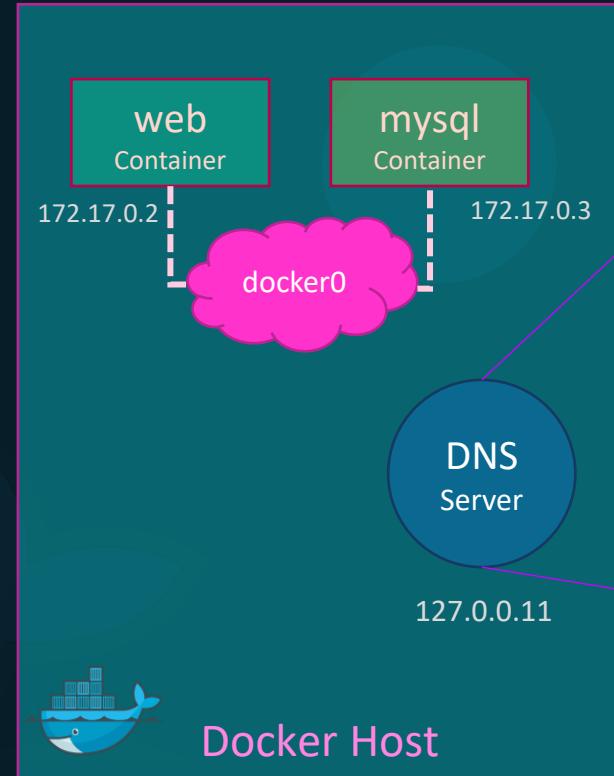
NETWORK ID	NAME	DRIVER	SCOPE
dba0fb9370fe	bridge	bridge	local
46d476b87cd9	customer-isolated-network	bridge	local
6de685cec1ce	docker_gwbridge	bridge	local
e29d188b4e47	host	host	local
mmrho7vsb9rm	ingress	overlay	swarm
d9f11695f0d6	none	null	local
d371b4009142	simplewebappdocker_default	bridge	local

Network Inspect



Embedded DNS

```
mysql.connect( mysql )
```



Host	IP
web	172.17.0.2
mysql	172.17.0.3

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pbUu>



KodeKloud

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pJUu>



d o c k e r registry

Image

```
▶ docker run nginx
```

Image

`image: nginx`



Image/
Repository

Image

image: nginx/nginx



User/ Image/
Account Repository

Image

image: docker.io/nginx/nginx

Registry User/ Image/
Account Repository

gcr.io/kubernetes-e2e-test-images/dnsutils

Private Registry

```
▶ docker login private-registry.io
```

Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to <https://hub.docker.com> to create one.

Username: registry-user

Password:

WARNING! Your password will be stored unencrypted in `/home/vagrant/.docker/config.json`.

Login Succeeded

```
▶ docker run private-registry.io/apps/internal-app
```

Deploy Private Registry

```
▶ docker run -d -p 5000:5000 --name registry registry:2
```

```
▶ docker image tag my-image localhost:5000/my-image
```

```
▶ docker push localhost:5000/my-image
```

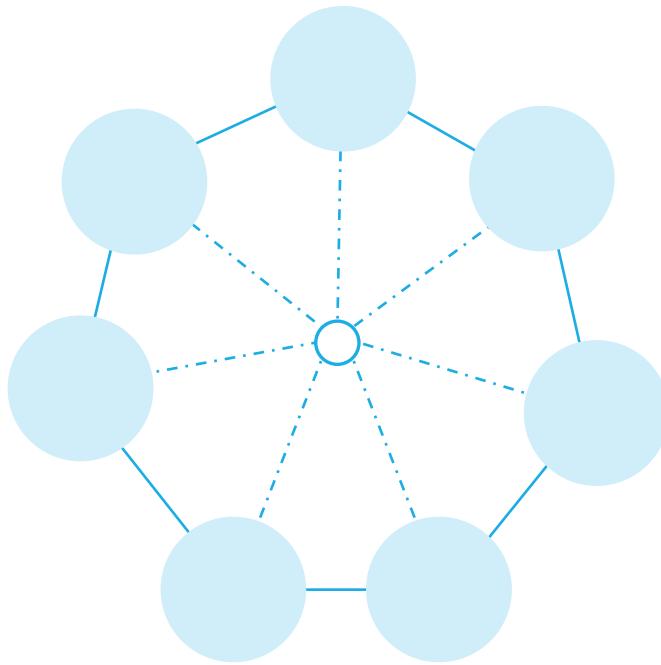
```
▶ docker pull localhost:5000/my-image
```

```
▶ docker pull 192.168.56.100:5000/my-image
```



KodeKloud

Check out our full course on Docker for the Absolute Beginners: <https://kode.wiki/3N9pJUu>



kubernetes

for beginners

Who am I?



I am an IT Solutions Architect focusing on Cloud automation and DevOps. I am passionate about learning new technology and teaching. I believe the best way to learn is to learn by doing and in a fun way. I have authored multiple courses on DevOps and cloud automation technologies. My courses focus on providing students with an interactive and hands-on experience in learning new technology that makes learning really interesting.

Total Students
17,375

Courses
5

Reviews
1,849

Courses you are teaching



Docker - SWARM |
SERVICES | STACKS - ...

Mumshad Mannambeth

★★★★★ 4.6 (81)

\$199.99 \$11.99

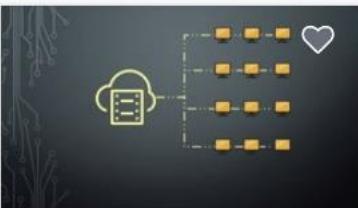


Docker for the Absolute
Beginner - Hands On

Mumshad Mannambeth

★★★★★ 4.5 (564)

\$64.99 \$11.99

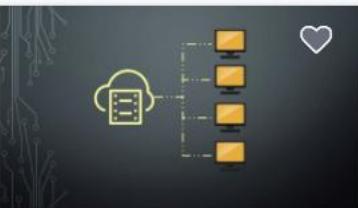


Ansible Advanced

Mumshad Mannambeth

★★★★★ 4.4 (152)

\$154.99 \$11.99



Ansible for the Absolute
Beginner - Hands-On

Mumshad Mannambeth

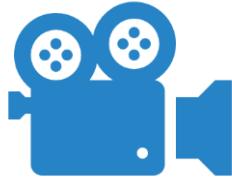
★★★★★ 4.5 (991)

\$99.99 \$11.99

Course Structure



Lecture



Demos



Quiz



Lab Exercises



Tips &
Tricks



Q&A

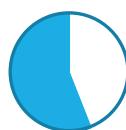
Who is this for?

Developers

System Admins

Managers

How to attend this course?



Lectures



Demos



Quizzes



Lab Exercises



Labs

Objectives

- Kubernetes Overview
- Containers – Docker
- Container Orchestration?
- Demo - Setup Kubernetes
- Kubernetes Concepts – PODs | ReplicaSets | Deployment | Services
- Networking in Kubernetes
- Kubernetes Management - Kubectl
- Kubernetes Definition Files - YAML
- Kubernetes on Cloud – AWS/GCP

kubernetes or K8s

Container + Orchestration



KodeKloud

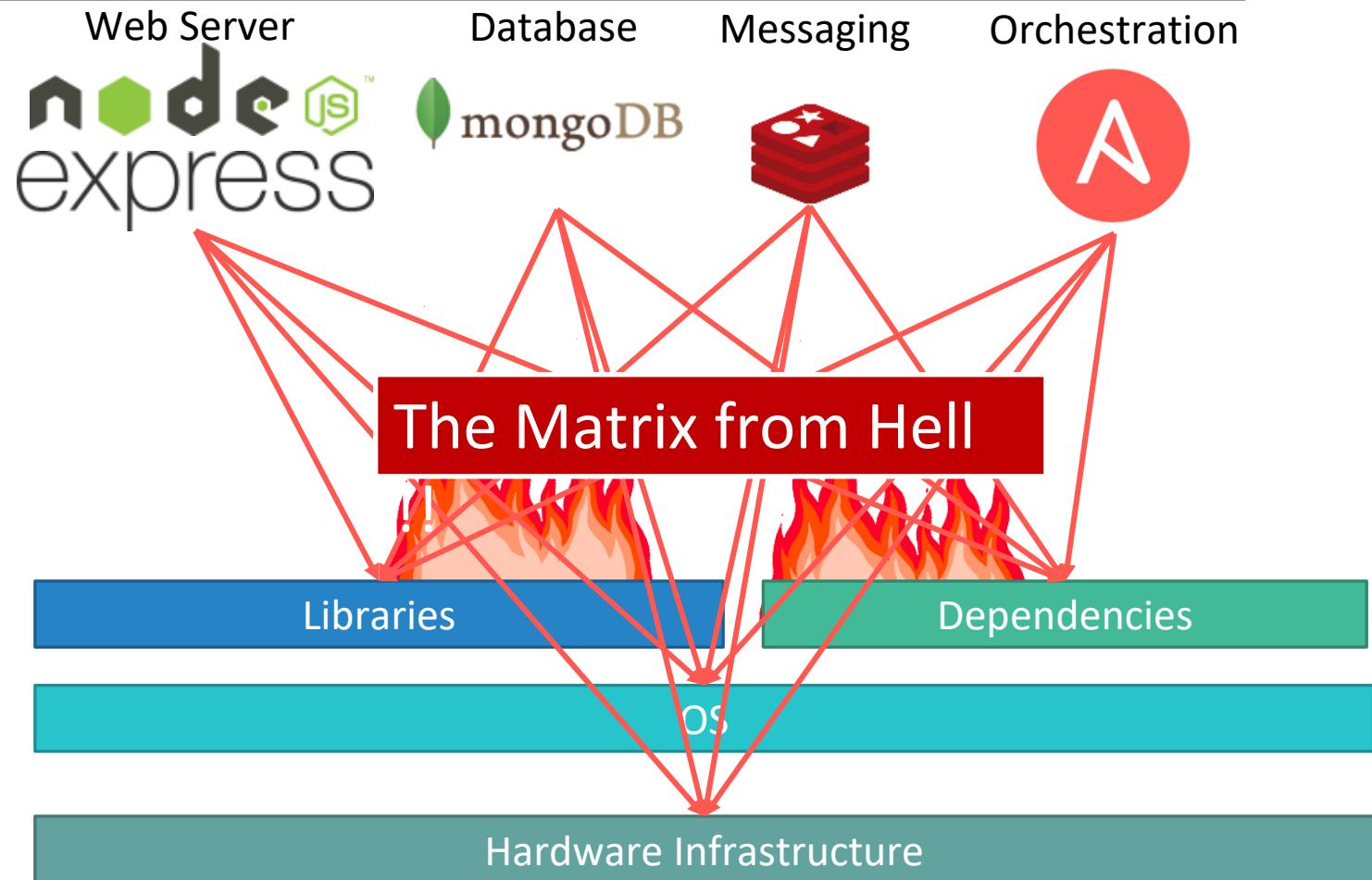
Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

Containers

mumshad mannambeth

Why do you need containers?

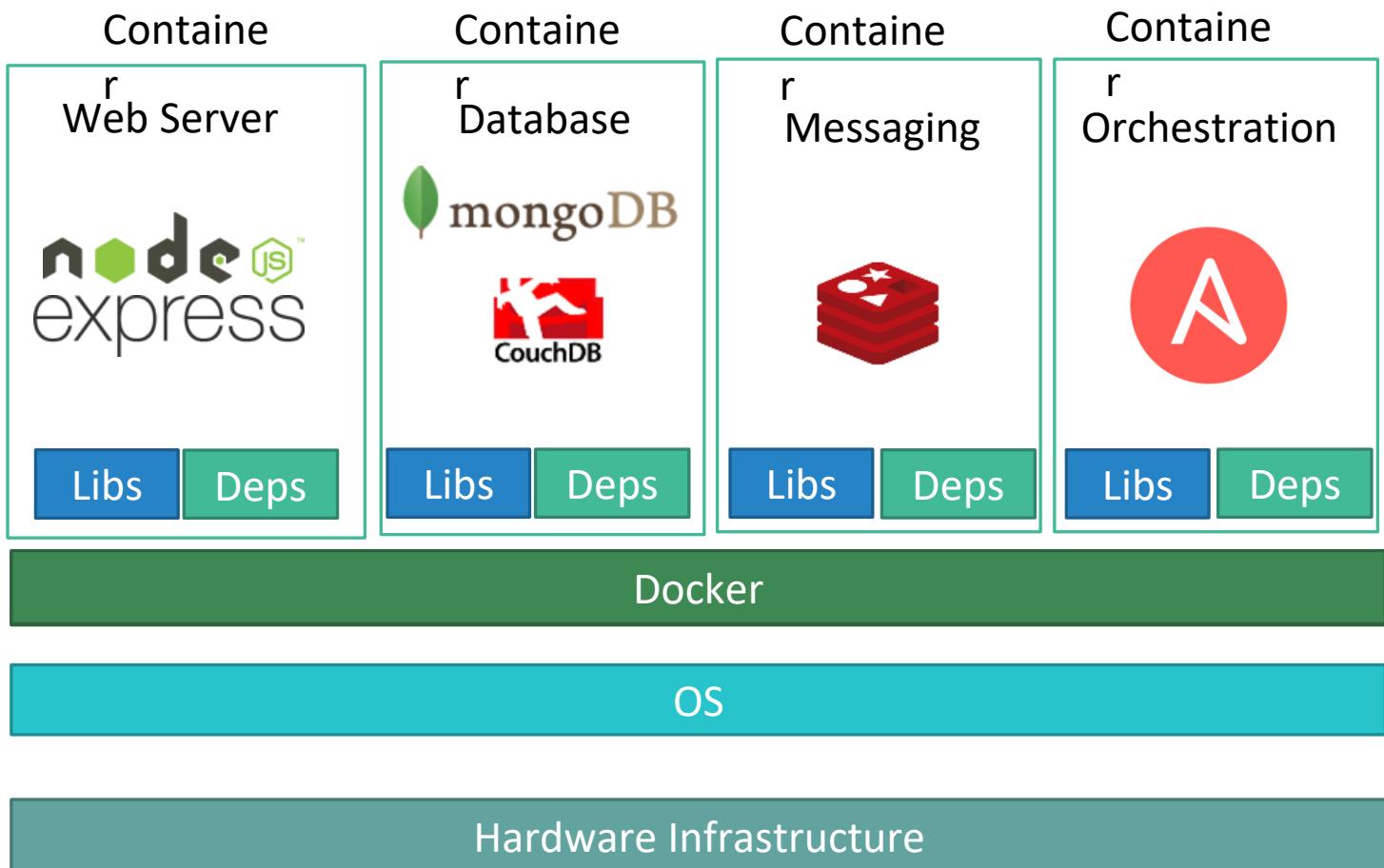
- Compatibility/Dependency
- Long setup time
- Different Dev/Test/Prod environments



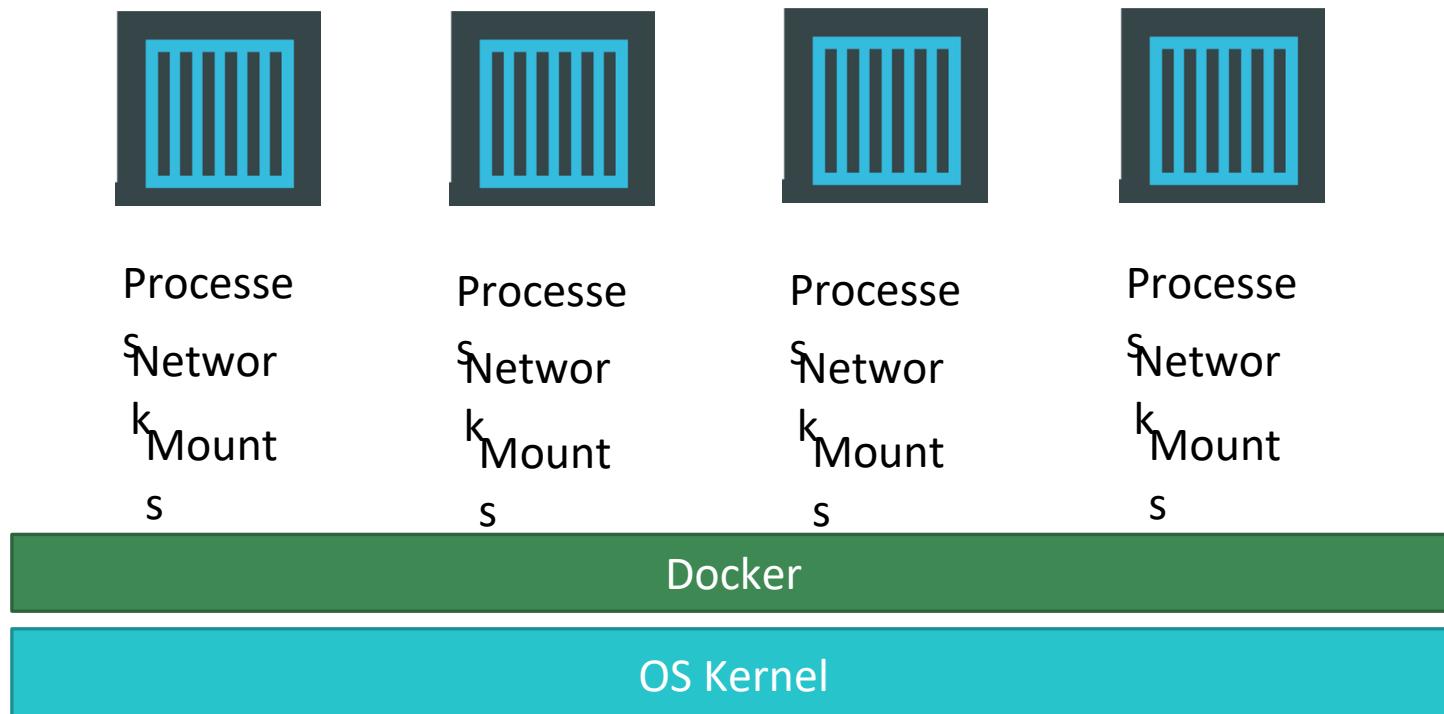
What can it do?

Containerize Applications

Run each service with its own
dependencies in separate containers



What are containers?



Operating system



OS

Software

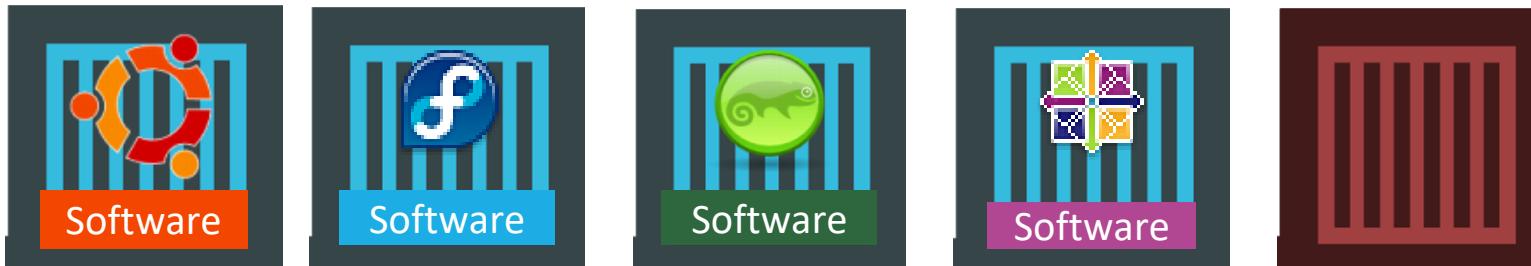
Software

Software

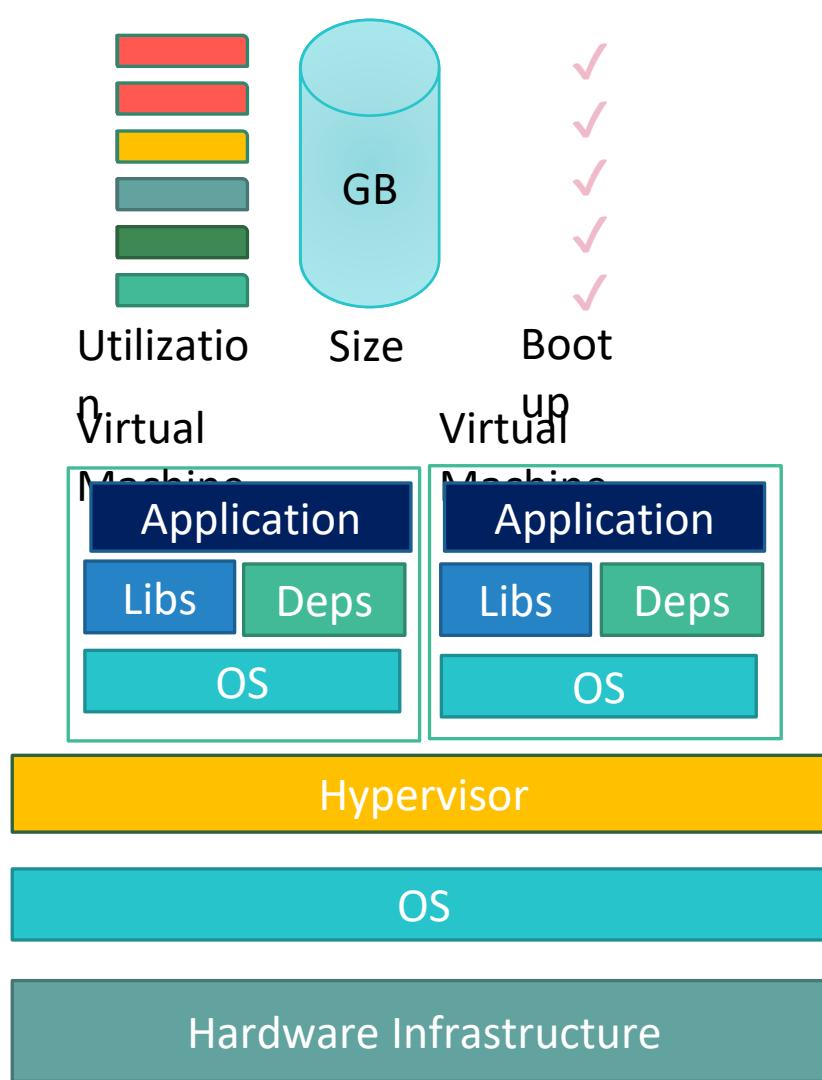
Software

OS Kernel

Sharing the kernel

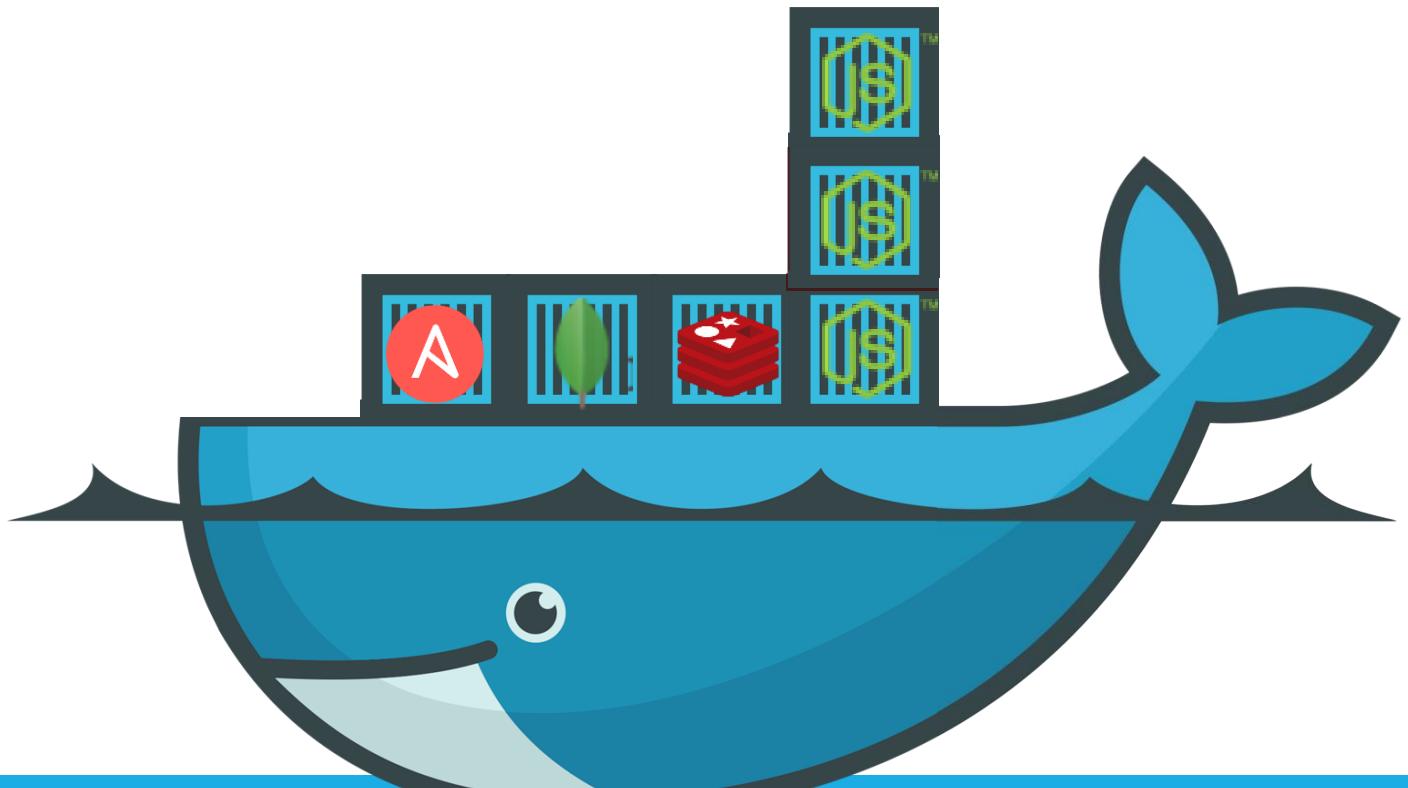


Containers vs Virtual Machines



How is it done?

```
docker run ansible  
docker run mongodb  
docker run redis  
docker run nodejs  
docker run nodejs  
docker run nodejs
```



Container vs image



Docker
Image
Package
Templat
e
Plan



Docker Container #1



Docker Container #2



Docker Container #3

Container Advantage

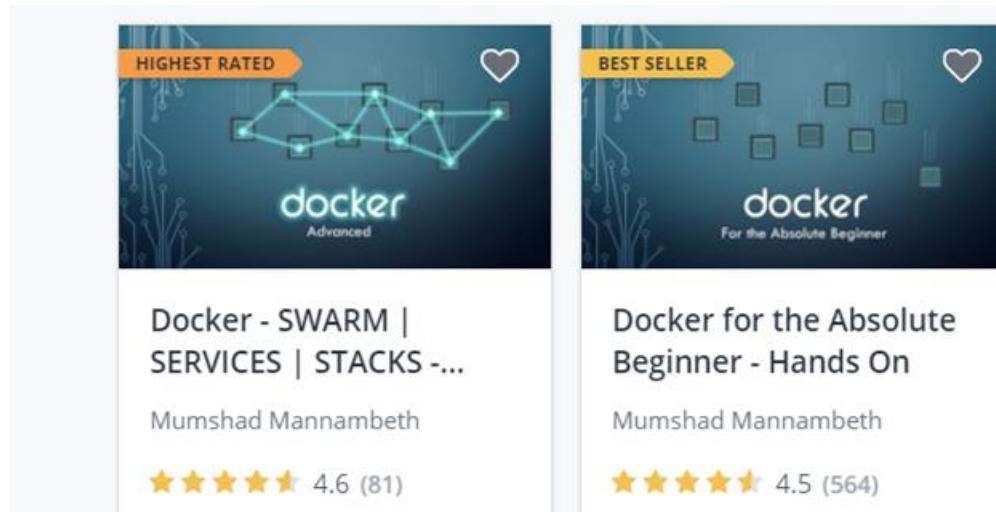


Developer



Operations

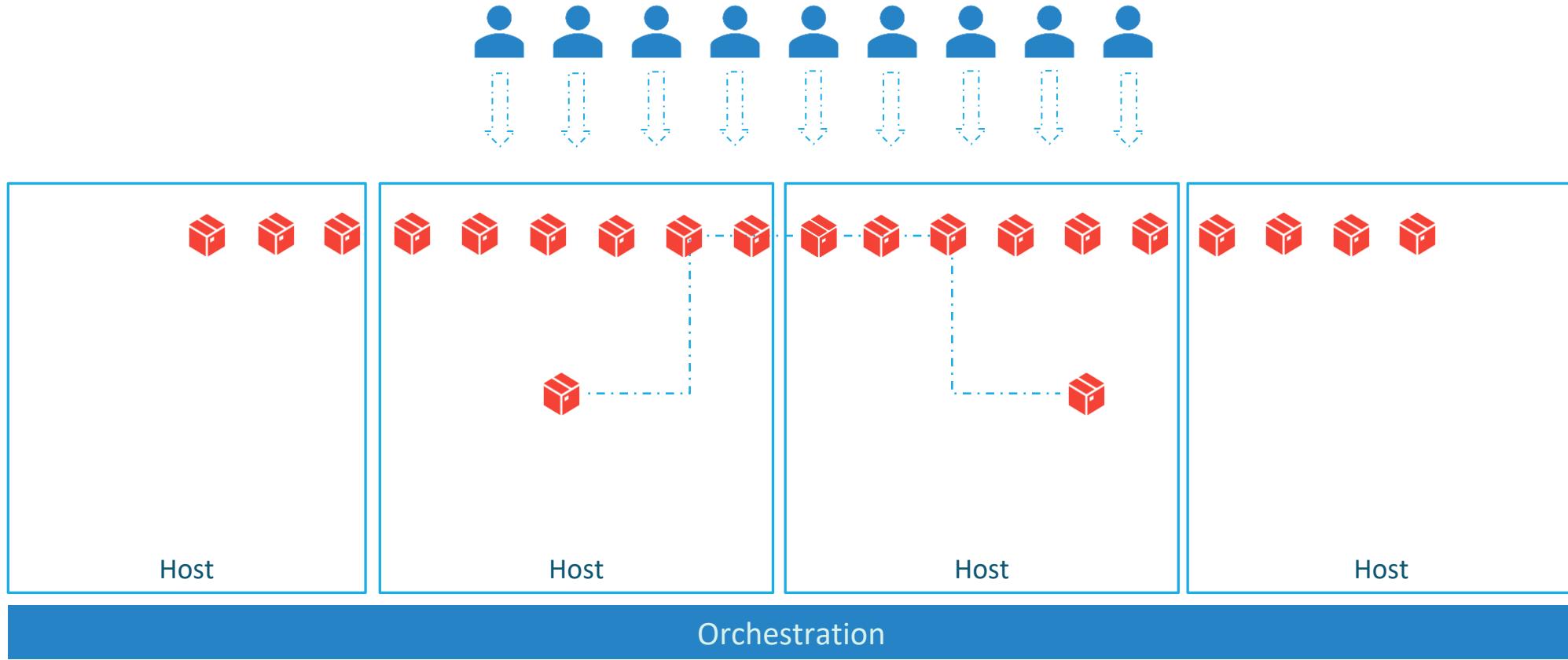
More about containers



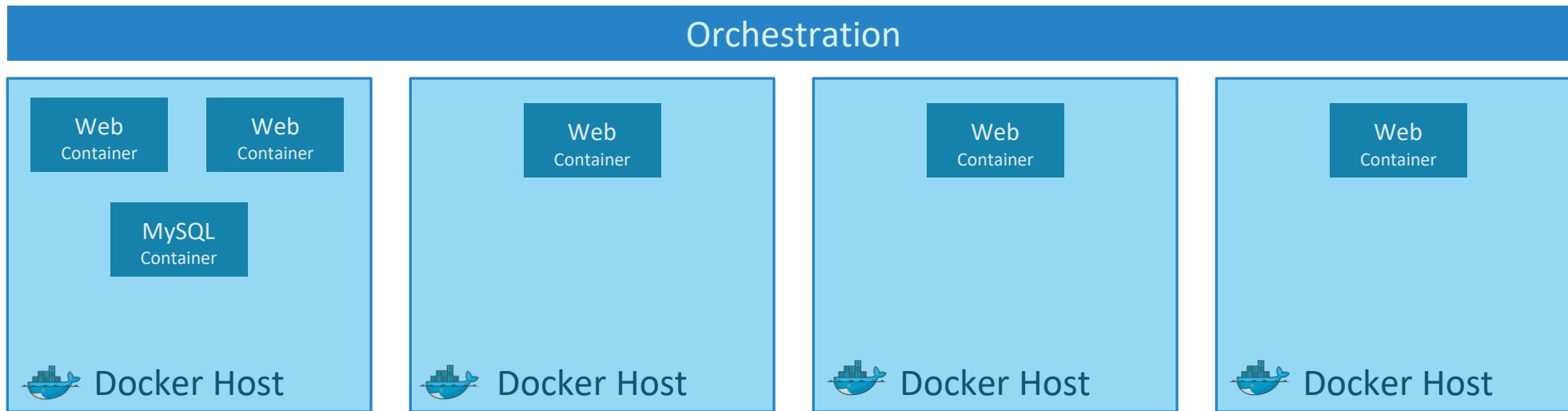
Container Orchestration

mumshad mannambeth

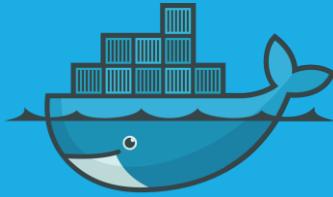
Container Orchestration



Container orchestration



Orchestration Technologies



Docker Swarm

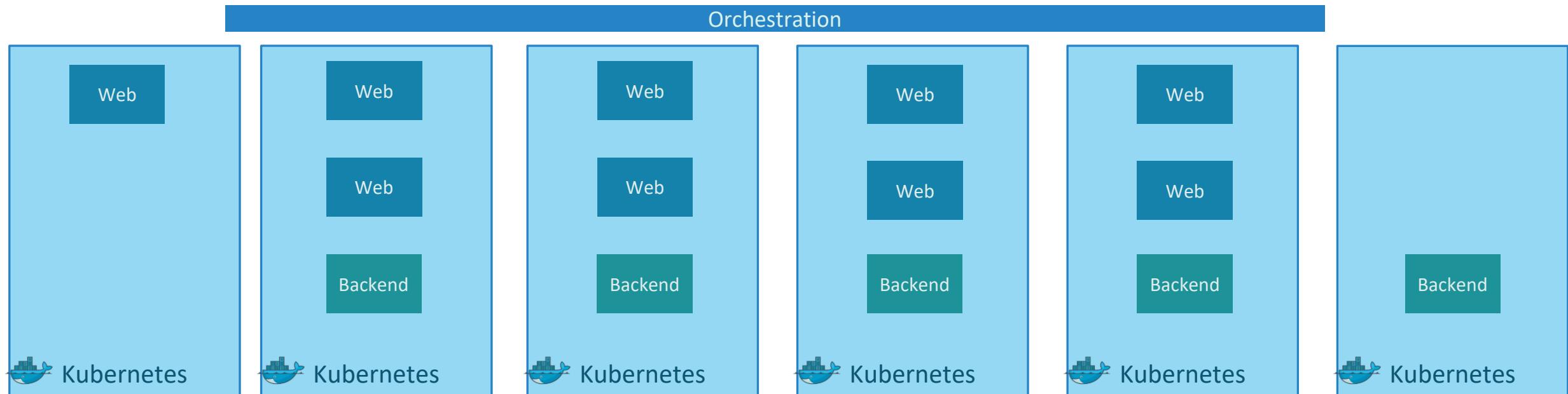


kubernetes



MESOS

Kubernetes Advantage



And that is **kubernetes..**



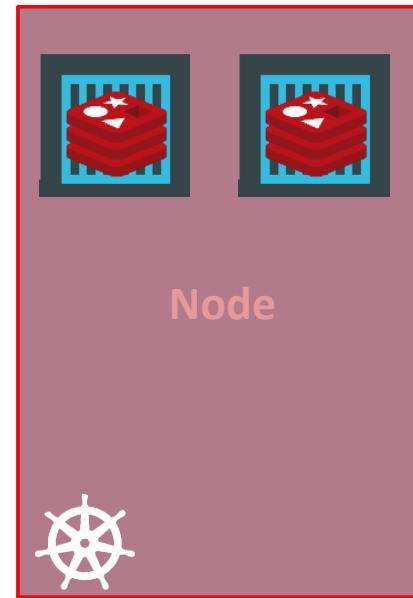
KodeKloud

Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

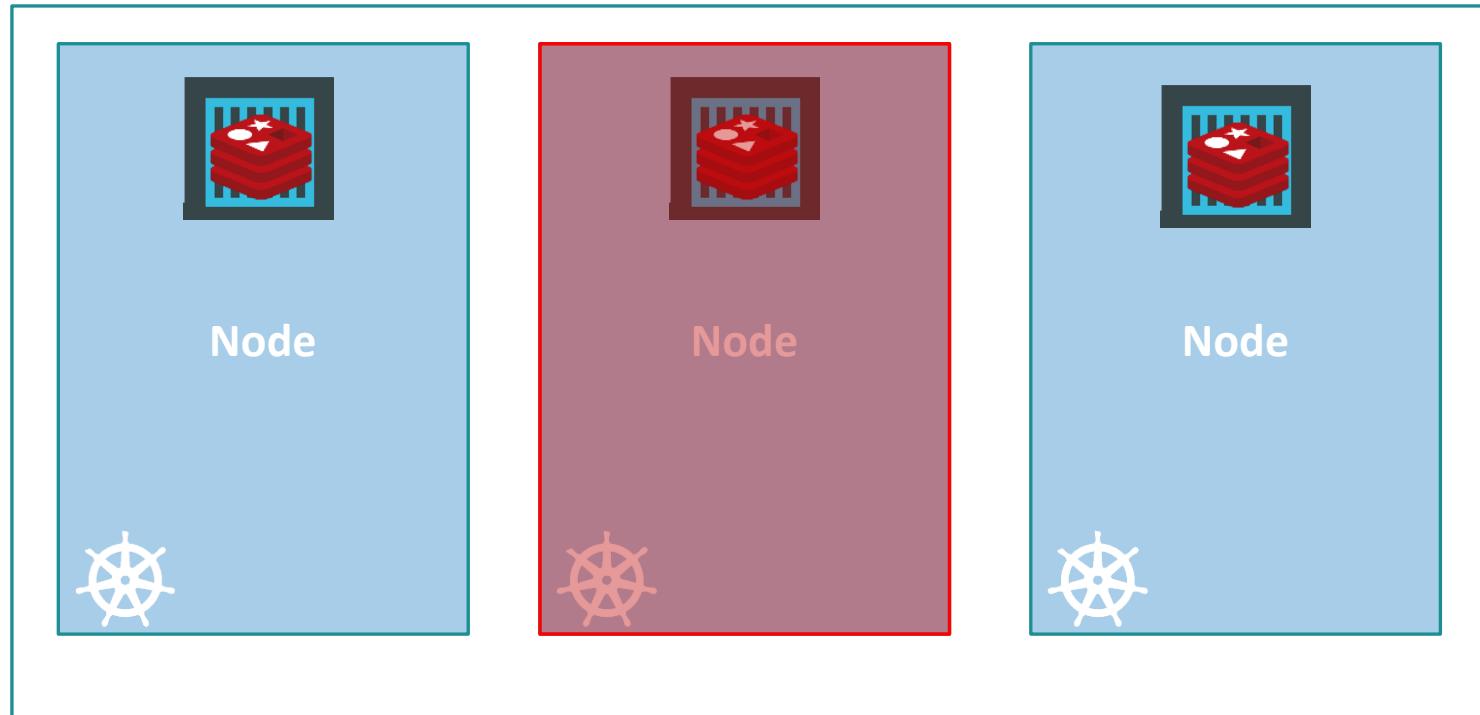
Architecture

mumshad mannambeth

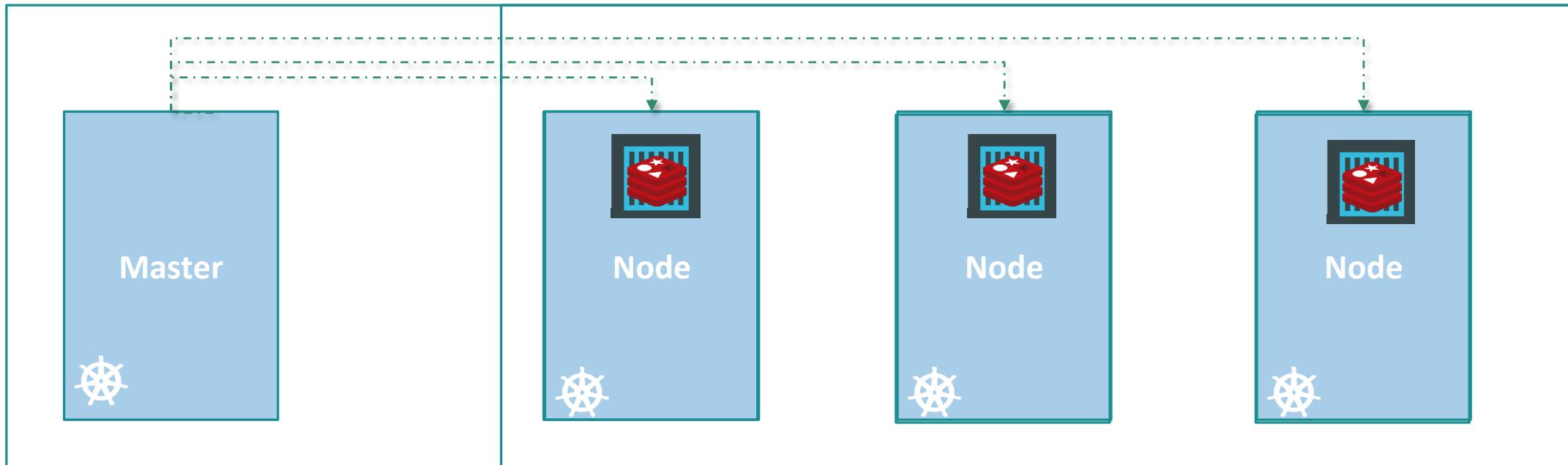
Nodes(Minions)



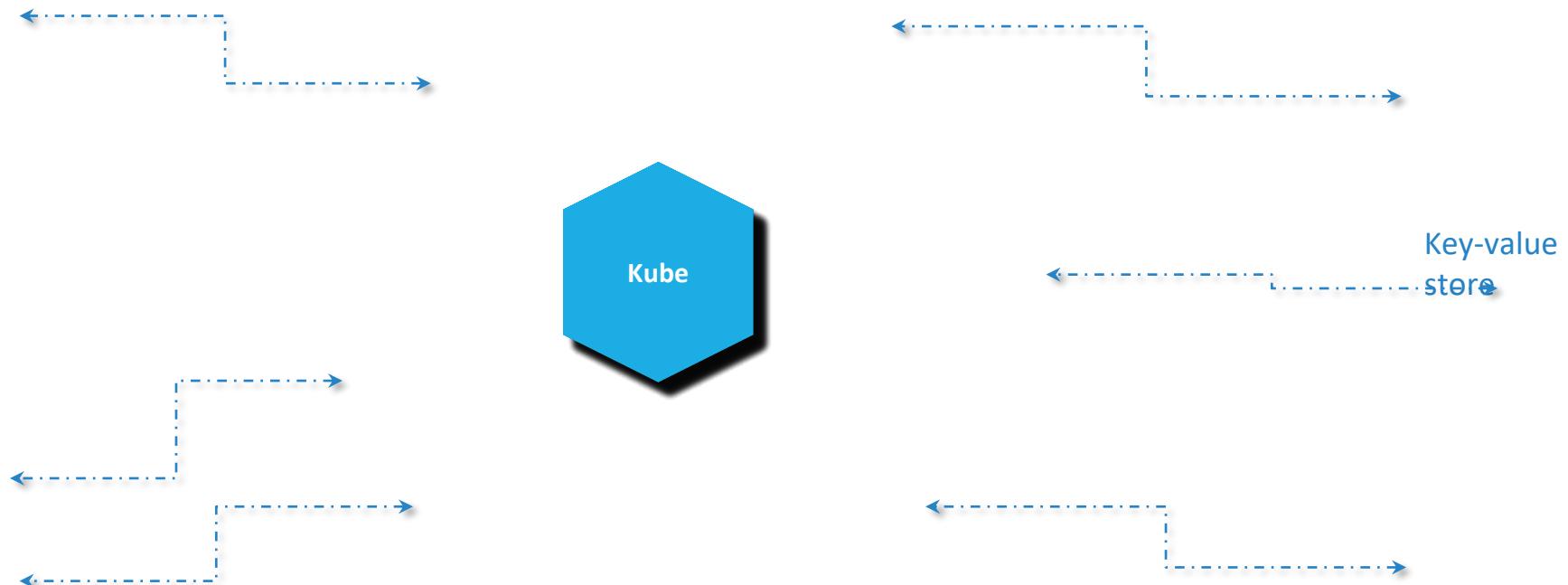
Cluster



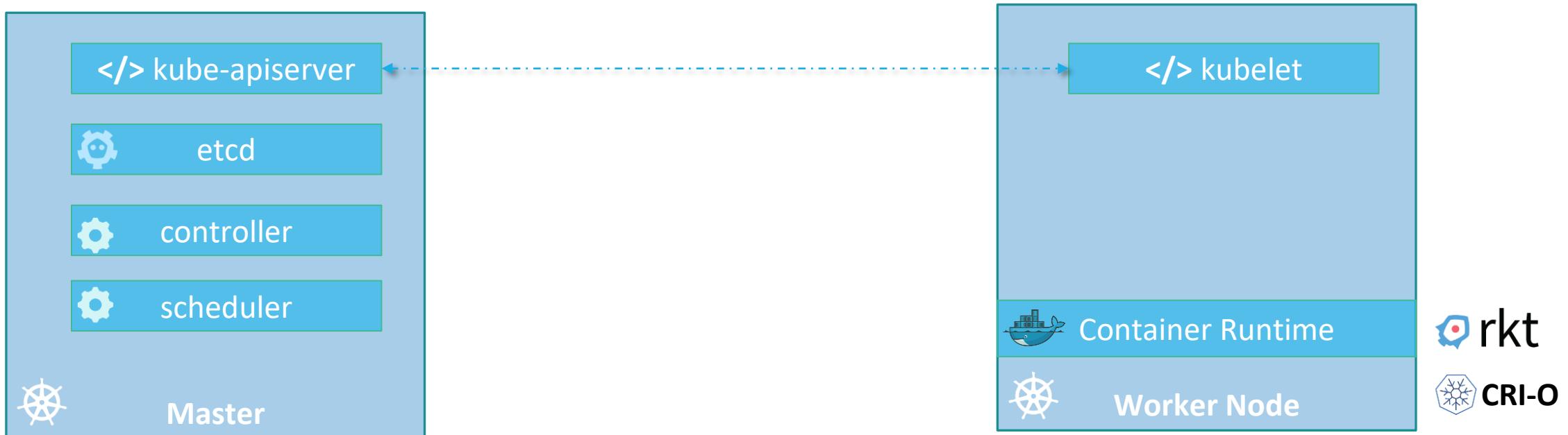
Master



Components



Master vs Worker Nodes

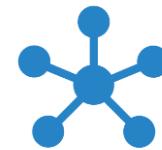


kubectl

```
kubectl run hello-minikube
```



```
kubectl cluster-info
```



```
kubectl get nodes
```





KodeKloud

Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

Setup

mumshad mannambeth

Setup Kubernetes



Minikube



MicroK8s



Kubeadm



Google Cloud Platform



Amazon Web Services

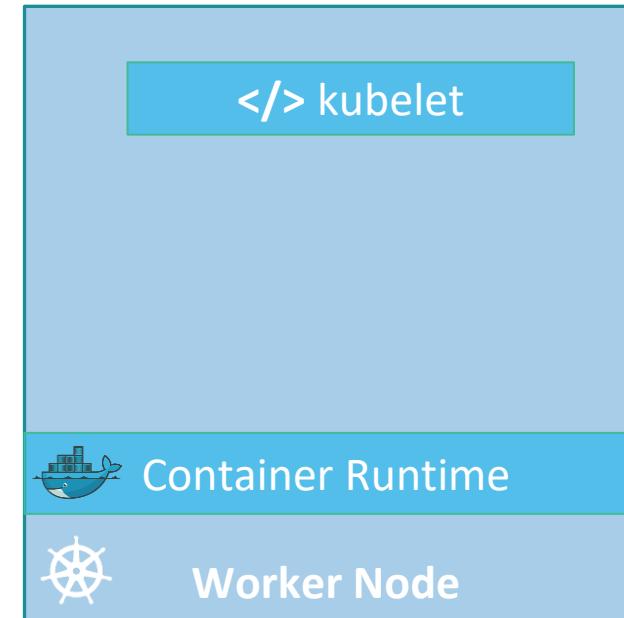
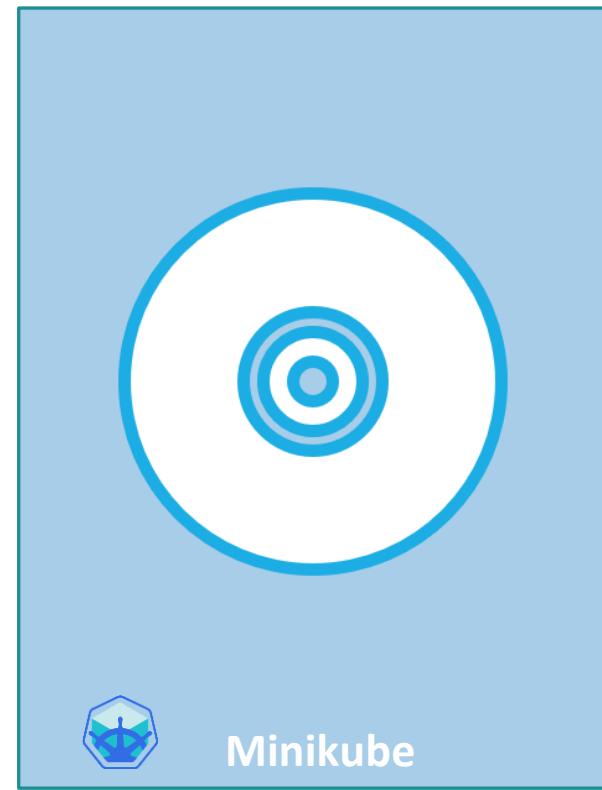
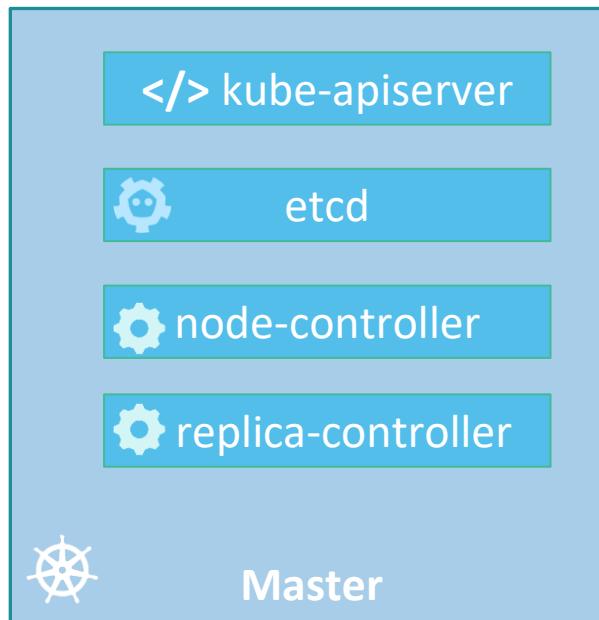


Microsoft Azure

play-with-k8s.com

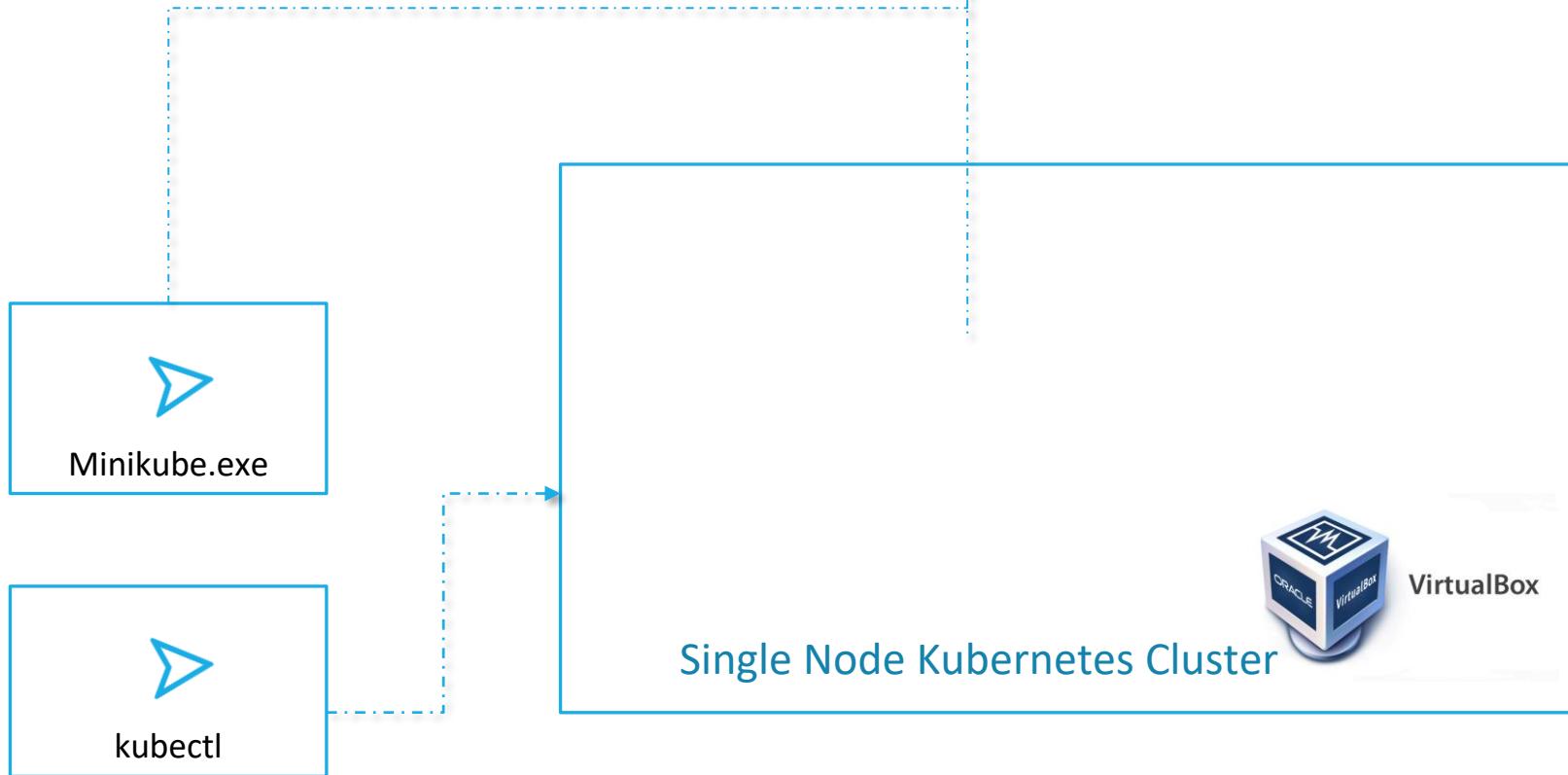


Minikube





Minikube



Demo

minikube

POD

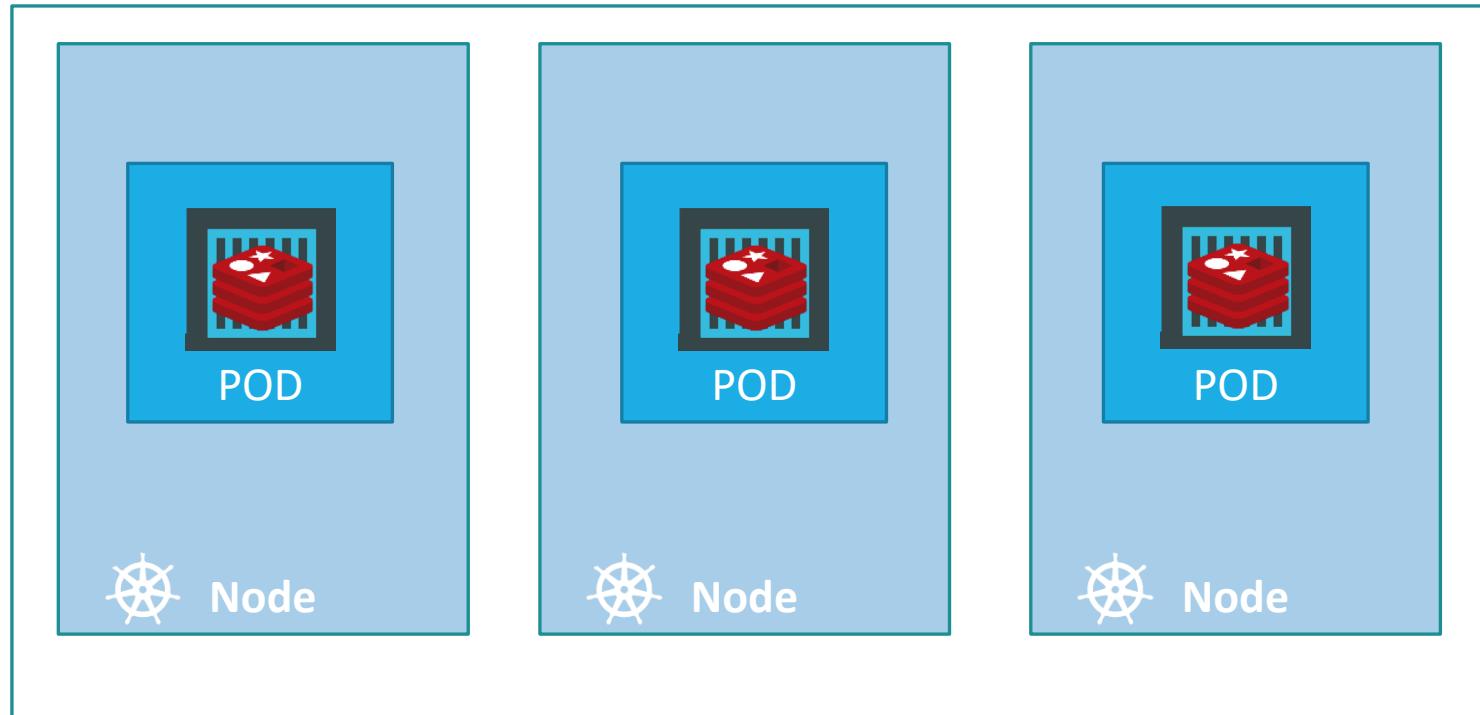
mumshad mannambeth

Assumptions

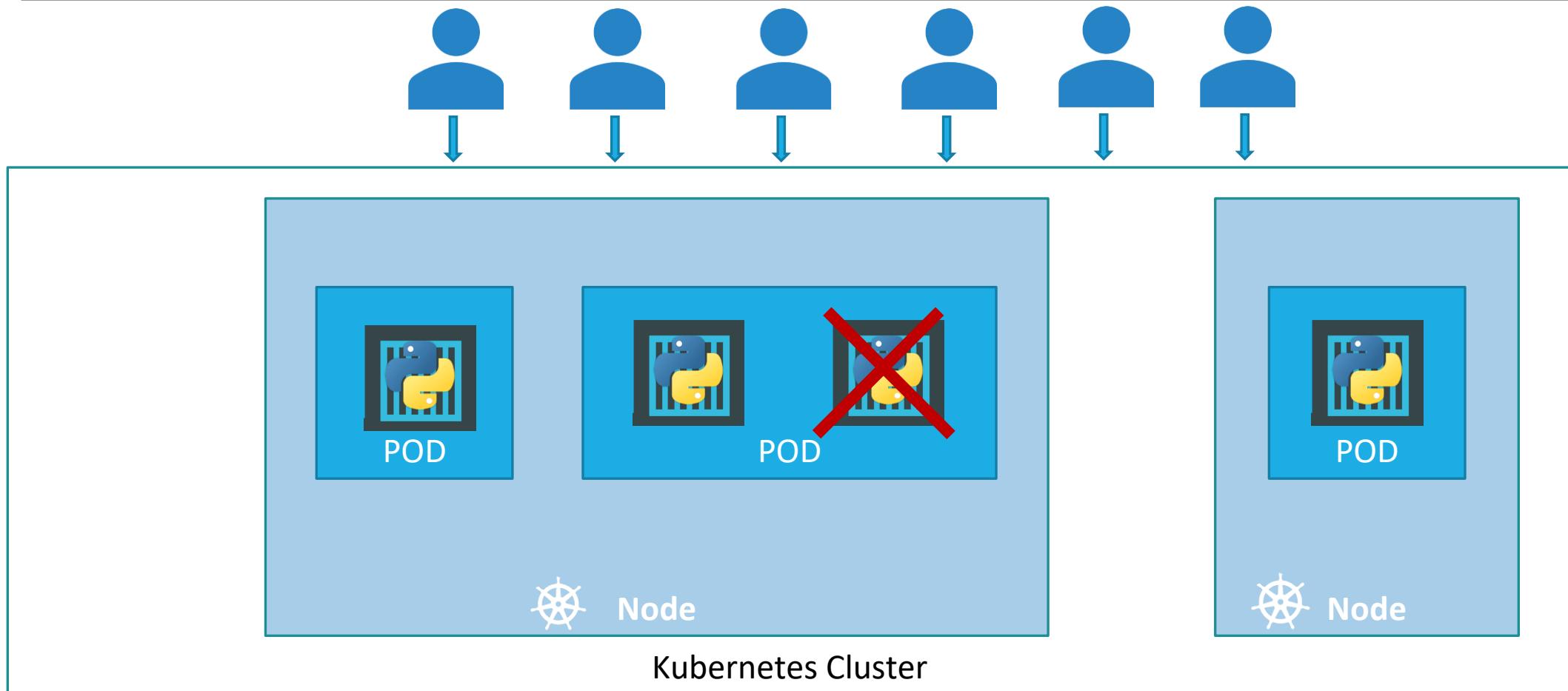
Docker Image

Kubernetes Cluster

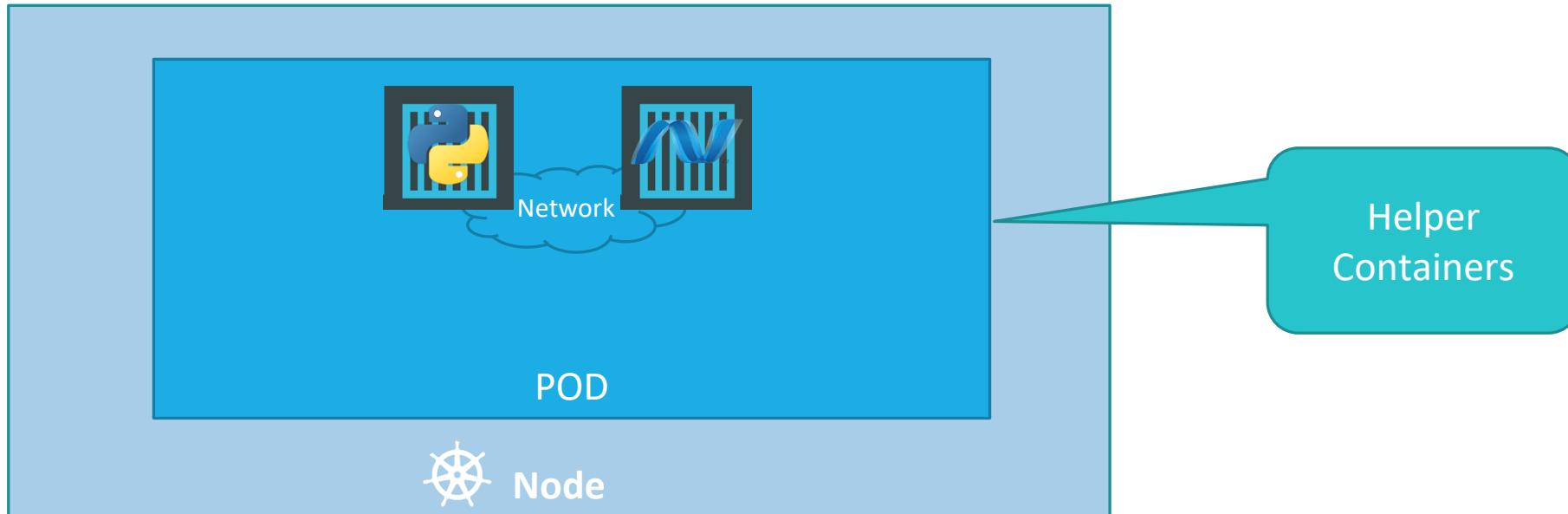
POD



POD



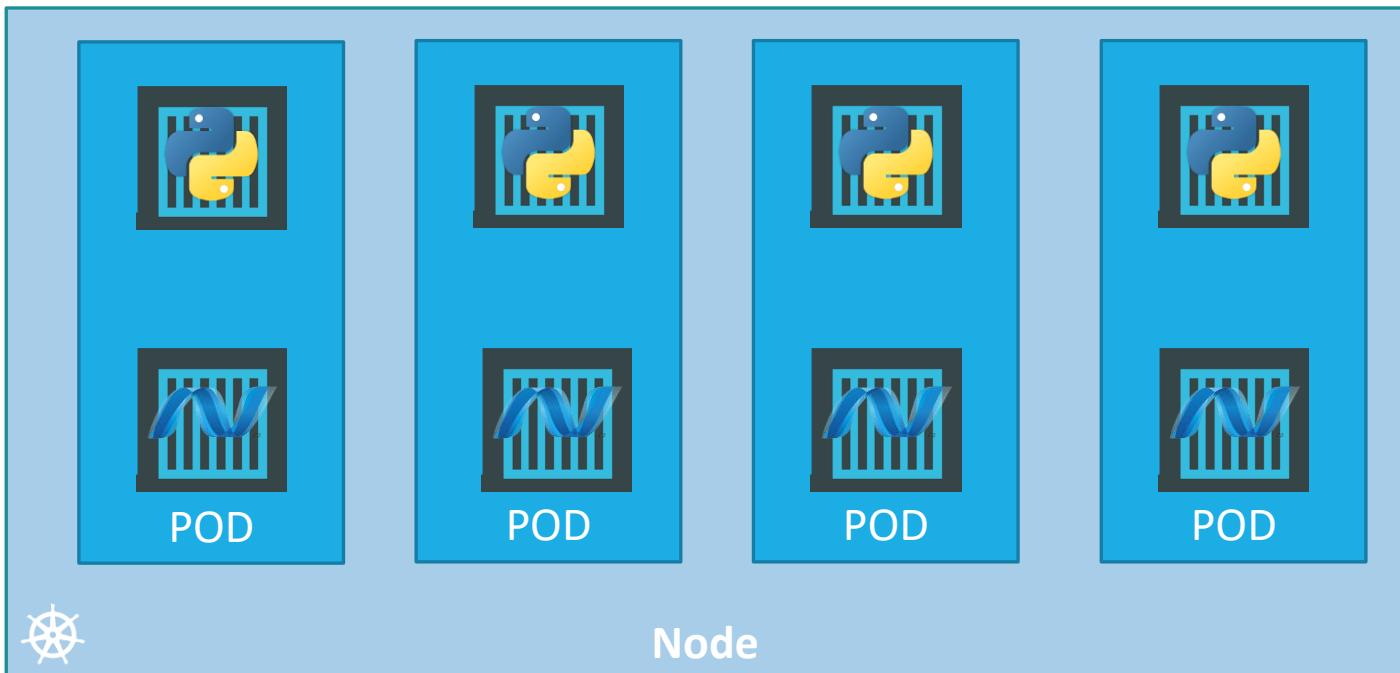
Multi-Container PODs



PODs Again!

```
docker run python-app  
docker run python-app  
docker run python-app  
docker run python-app  
  
docker run helper -link app1  
docker run helper -link app2  
docker run helper -link app3  
  
docker run helper -link app4
```

App	Helper	Volume
Python 1	App1	Vol1
Python 2	App2	Vol2



Note: I am avoiding networking and load balancing details to keep explanation simple.

Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>



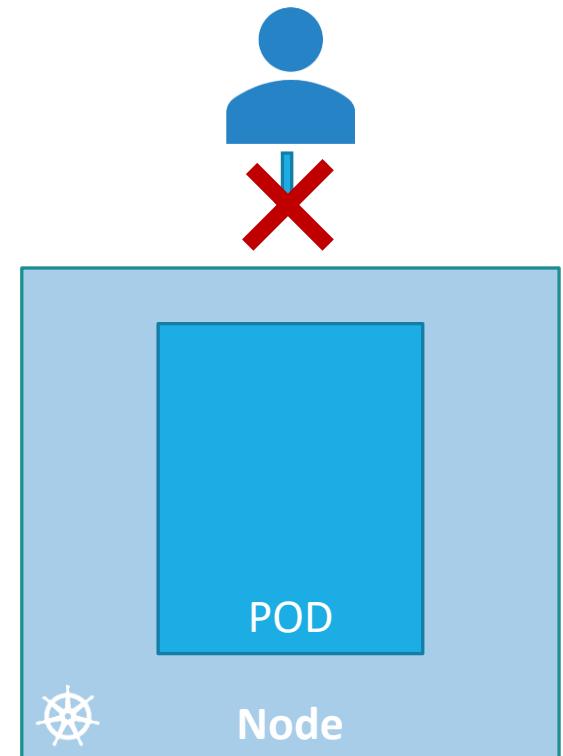
kubectl

```
kubectl run nginx --image nginx
```

```
kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx	0/1	ContainerCreating	0	6s

NAME	READY	STATUS	RESTARTS	AGE
nginx	1/1	Running	0	34s



Demo

POD



KodeKloud

Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

YAML

WHAT IS YAML?

XML

```
<Servers>
  <Server>
    <name>Server1</name>
    <owner>John</owner>
    <created>12232012</created>
    <status>active</status>
  </Server>
</Servers>
```

JSON

```
{
  Servers: [
    {
      name: Server1,
      owner: John,
      created: 12232012,
      status: active,
    }
  ]
}
```

YAML

```
Servers:
  - name: Server1
    owner: John
    created: 12232012
    status: active
```



WHAT IS YAML?

XML

```
<Servers>
  <Server>
    <name>Server1</name>
    <owner>John</owner>
    <created>12232012</created>
    <status>active</status>
  </Server>
</Servers>
```

JSON

```
{
  Servers: [
    {
      name: Server1,
      owner: John,
      created: 12232012,
      status: active,
    }
  ]
}
```

YAML

```
Servers:
  - name: Server1
    owner: John
    created: 12232012
    status: active
```



YAML - NOTES

Dictionary/Map

```
Banana:  
    Calories: 105  
    Fat: 0.4 g  
    Carbs: 27 g
```



```
Banana:  
    Calories: 105  
    Carbs: 27 g  
    Fat: 0.4 g
```



Dictionary – Unordered
List – Ordered

Array/List

```
Fruits:  
- Orange  
- Apple  
- Banana
```



```
Fruits:  
- Orange  
- Banana  
- Apple
```

```
# List of Fruits  
Fruits:  
- Orange  
- Apple  
- Banana
```



Hash # – Comments



SPACES

Press **Esc** to exit full screen

Dictionary/Map



Equal number of spaces



Banana:
Calories: 105
Fat: 0.4 g
Carbs: 27 g

Fat
.4

Cal
105

Crb
27



YAML - ADVANCED

Press `Esc` to exit full screen

Key Value/Dictionary/Lists

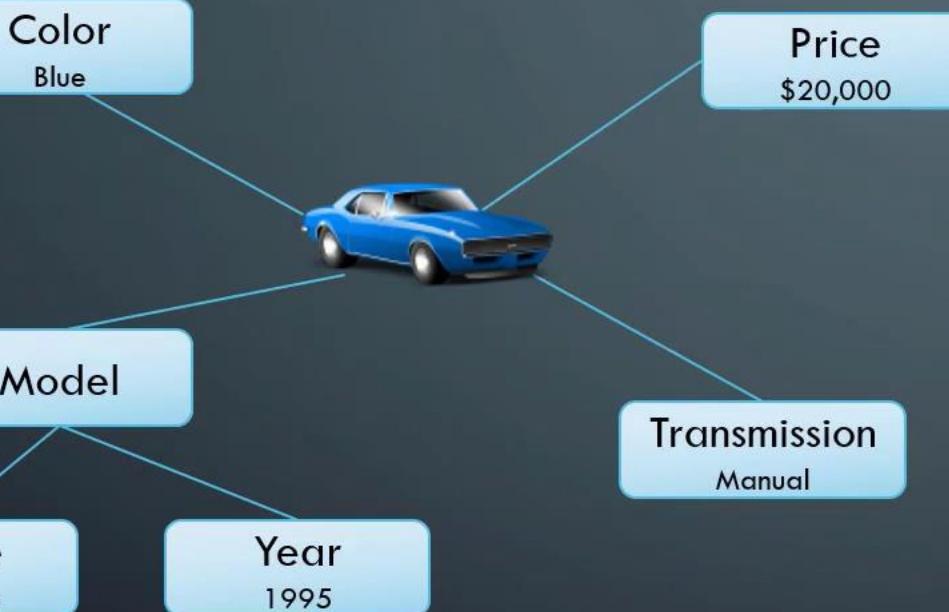
Fruits:

- Banana:
 - Calories: 105
 - Fat: 0.4 g
 - Carbs: 27 g

- Grape:
 - Calories: 62
 - Fat: 0.3 g
 - Carbs: 16 g



DICTIONARY vs LIST vs LIST OF DICTIONARIES



Dictionary In Dictionary

```
Color: Blue  
Model:  
  Name: Corvette  
  Year: 1995  
Transmission: Manual  
Price: $20,000
```



DICTIONARY vs LIST vs LIST OF DICTIONARIES



Color: Blue
Model:
Name: Corvette
Model: 1995
Transmission: Manual
Price: \$20,000



Color: Grey
Model:
Name: Corvette
Model: 1995
Transmission: Manual
Price: \$22,000



Color: Red
Model:
Name: Corvette
Model: 1995
Transmission: Automatic
Price: \$20,000



Color: Green
Model:
Name: Corvette
Model: 1995
Transmission: Manual
Price: \$23,000



Color: Blue
Model:
Name: Corvette
Model: 1995
Transmission: Manual
Price: \$20,000



Color: Black
Model:
Name: Corvette
Model: 1995
Transmission: Automatic
Price: \$25,000

List Of Dictionaries

- Color: Blue
Model:
Name: Corvette
Model: 1995
Transmission : Manual
Price: \$20,000
- Color: Grey
Model:
Name: Corvette
Model: 1995
Transmission: Manual
Price: \$22,000
- Color: Red
Model:
Name: Corvette
Model: 1995
Transmission : Automatic
Price: \$20,000
- Color: Green
Model:
Name: Corvette
Model: 1995
Transmission : Manual
Price: \$23,000
- Color: Blue
Model:
Name: Corvette
Model: 1995
Transmission: Automatic
Price: \$20,000
- Color: Black
Model:
Name: Corvette
Model: 1995
Transmission : Manual
Price: \$25,000



YAML - NOTES

Dictionary/Map

```
Banana:  
  Calories: 105  
  Fat: 0.4 g  
  Carbs: 27 g
```



```
Banana:  
  Calories: 105  
  Carbs: 27 g  
  Fat: 0.4 g
```



Dictionary – Unordered
List – Ordered

Array/List

```
Fruits:  
  - Orange  
  - Apple  
  - Banana
```



```
Fruits:  
  - Orange  
  - Banana  
  - Apple
```



Hash # – Comments

```
# List of Fruits  
Fruits:  
  - Orange  
  - Apple  
  - Banana
```



YAML in Kubernetes

```
pod-definition.yml
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
spec:
  containers:
    - name: nginx-container
      image: nginx
```

A large red 'X' is displayed in the center of the slide.

1st Item in List

```
kubectl create -f pod-definition.yml
```

Kind	Version
POD	v1
Service	v1
ReplicaSet	apps/v1
Deploymen t	apps/v1

YAML in Kubernetes

pod-definition.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
spec:
  containers:
    - name: nginx-container
      image: nginx
```

1st Item in List

```
kubectl create -f pod-definition.yml
```

Kind	Version
POD	v1
Service	v1
ReplicaSet	apps/v1
Deployment	apps/v1

Commands

```
> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
myapp-pod	1/1	Running	0	20s

```
> kubectl describe pod myapp-pod
```

```
Name:           myapp-pod
Namespace:      default
Node:          minikube/192.168.99.100
Start Time:    Sat, 03 Mar 2018 14:26:14 +0800
Labels:         app=myapp
                name=myapp-pod
Annotations:   <none>
Status:        Running
IP:            10.244.0.24
Containers:
  nginx:
    Container ID:  docker://830bb56c8c42a86b4bb70e9c1488fae1bc38663e4918b6c2f5a783e7688b8c9d
    Image:          nginx
    Image ID:      docker-pullable://nginx@sha256:4771d09578c7c6a65299e110b3ee1c0a2592f5ea2618d23e4ffe7a4cab1ce5de
    Port:          <none>
    State:         Running
      Started:    Sat, 03 Mar 2018 14:26:21 +0800
    Ready:         True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-x95w7 (ro)
Conditions:
  Type      Status
  Initialized  True
  Ready       True
  PodScheduled  True
Events:
  Type      Reason          Age   From            Message
  ----      ----          --   --              --
  Normal    Scheduled       34s   default-scheduler  Successfully assigned myapp-pod to minikube
  Normal    SuccessfulMountVolume 33s   kubelet, minikube  MountVolume.SetUp succeeded for volume "default-token-x95w7"
  Normal    Pulling          33s   kubelet, minikube  pulling image "nginx"
  Normal    Pulled           27s   kubelet, minikube  Successfully pulled image "nginx"
  Normal    Created          27s   kubelet, minikube  Created container
  Normal    Started          27s   kubelet, minikube  Started container
```



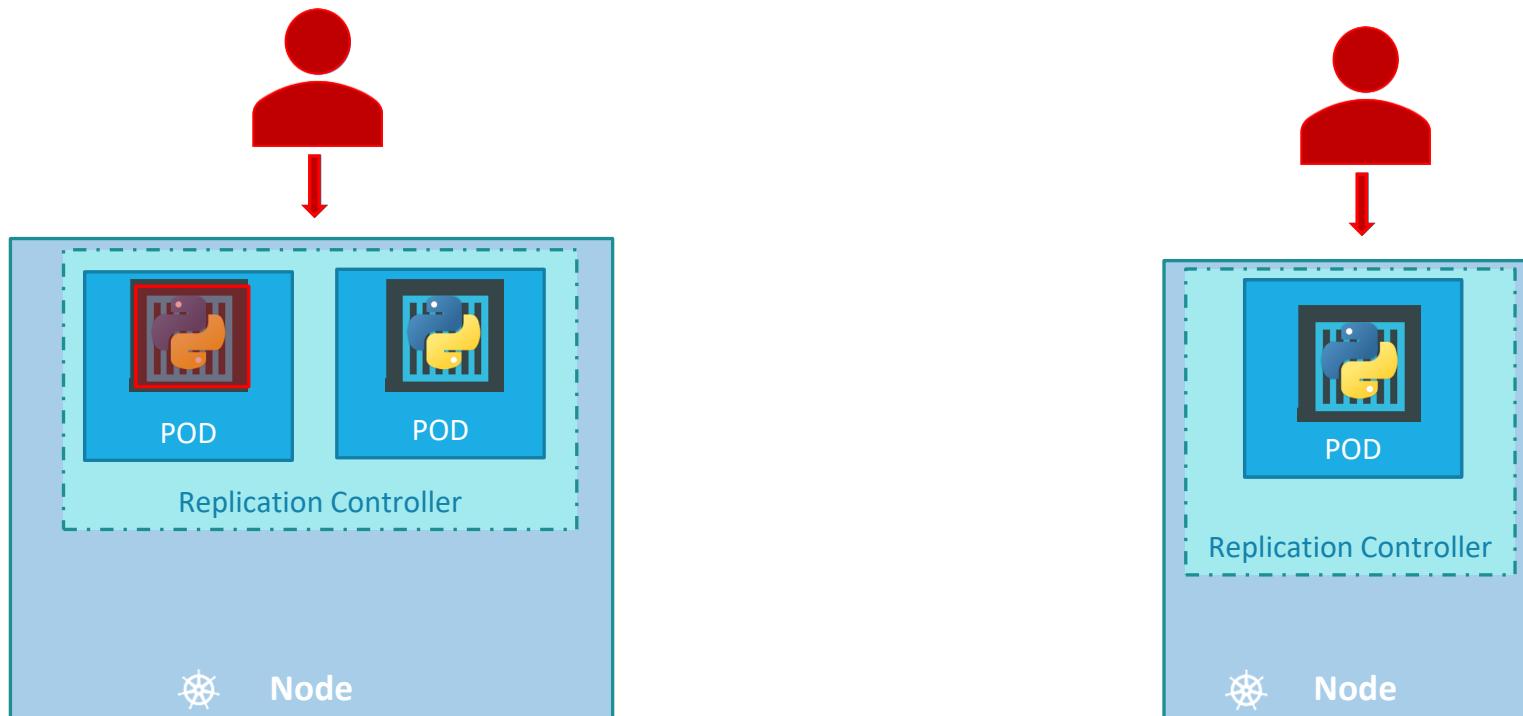
KodeKloud

Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

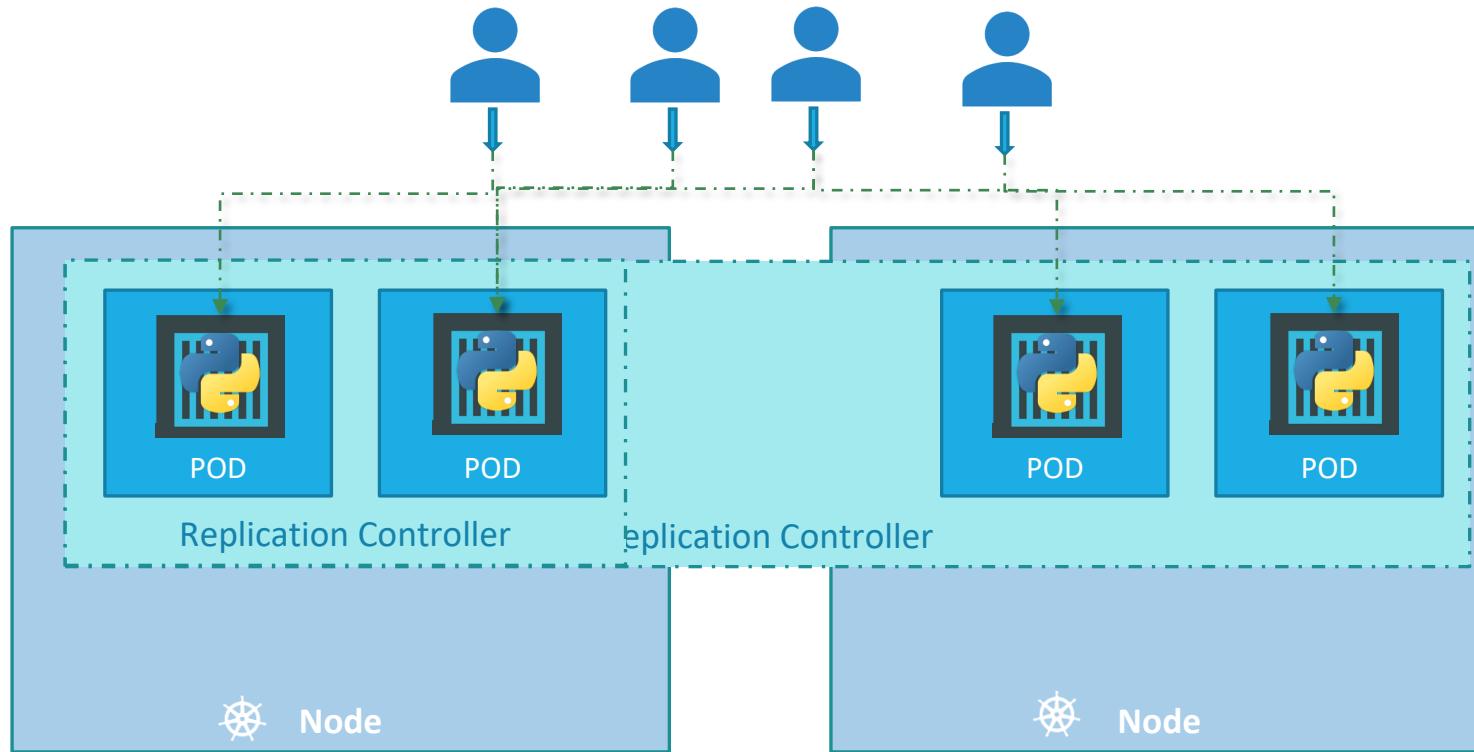
Replication Controller

mumshad mannambeth

High Availability



Load Balancing & Scaling

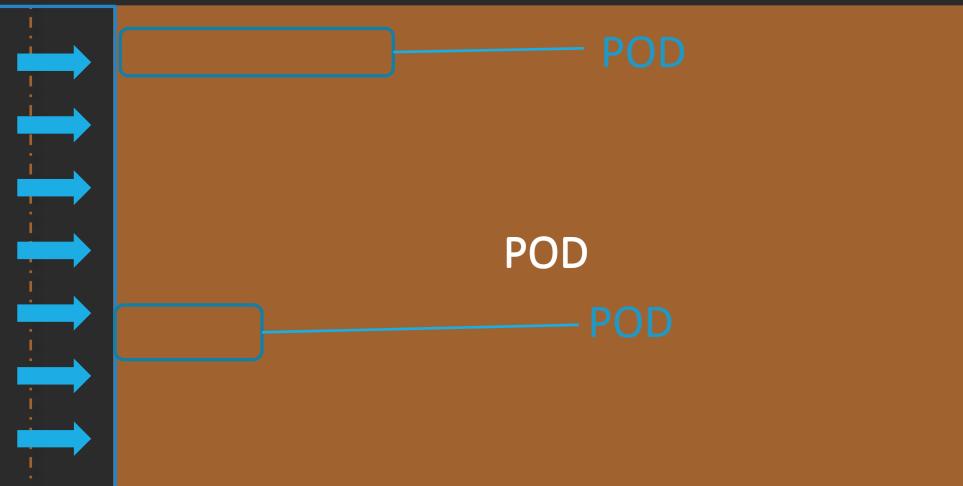


Replication Controller

Replica Set

`rc-definition.yml`

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: myapp-rc
  labels:
    app: myapp
    type: front-end
spec:
  template:
```



```
replicas: 3
```

`pod-definition.yml`

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
    type: front-end
spec:
  containers:
    - name: nginx-container
      image: nginx
```

```
> kubectl create -f rc-definition.yml
replicationcontroller "myapp-rc" created
```

```
> kubectl get replicationcontroller
NAME      DESIRED   CURRENT   READY   AGE
myapp-rc   3         3         3       19s
```

```
> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
myapp-rc-41vk9	1/1	Running	0	20s
myapp-rc-mc2mf	1/1	Running	0	20s
myapp-rc-px9pz	1/1	Running	0	20s

replicaset-definition.yml

```
apiVersion: apps/v1
```

```
kind: ReplicaSet
```

```
metadata:
```

```
  name: myapp-repl
```

```
  labels:
```

```
    app: myapp
```

```
    type: front-end
```

```
spec:
```

```
  template:
```

```
error: unable to recognize "replicaset-  
definition.yml": no matches for /, Kind=ReplicaSet
```

POD

```
  replicas: 3
```

```
  selector:
```

```
    matchLabels:
```

```
      type: front-end
```

pod-definition.yml

```
apiVersion: v1
```

```
kind: Pod
```

```
  labels:
```

```
    app: myapp
```

```
    type: front-end
```

```
  spec:
```

```
    containers:
```

```
      - name: nginx-container
```

```
        image: nginx
```

```
> kubectl create -f replicaset-definition.yml
```

```
replicaset "myapp-replicaset" created
```

```
> kubectl get replicaset
```

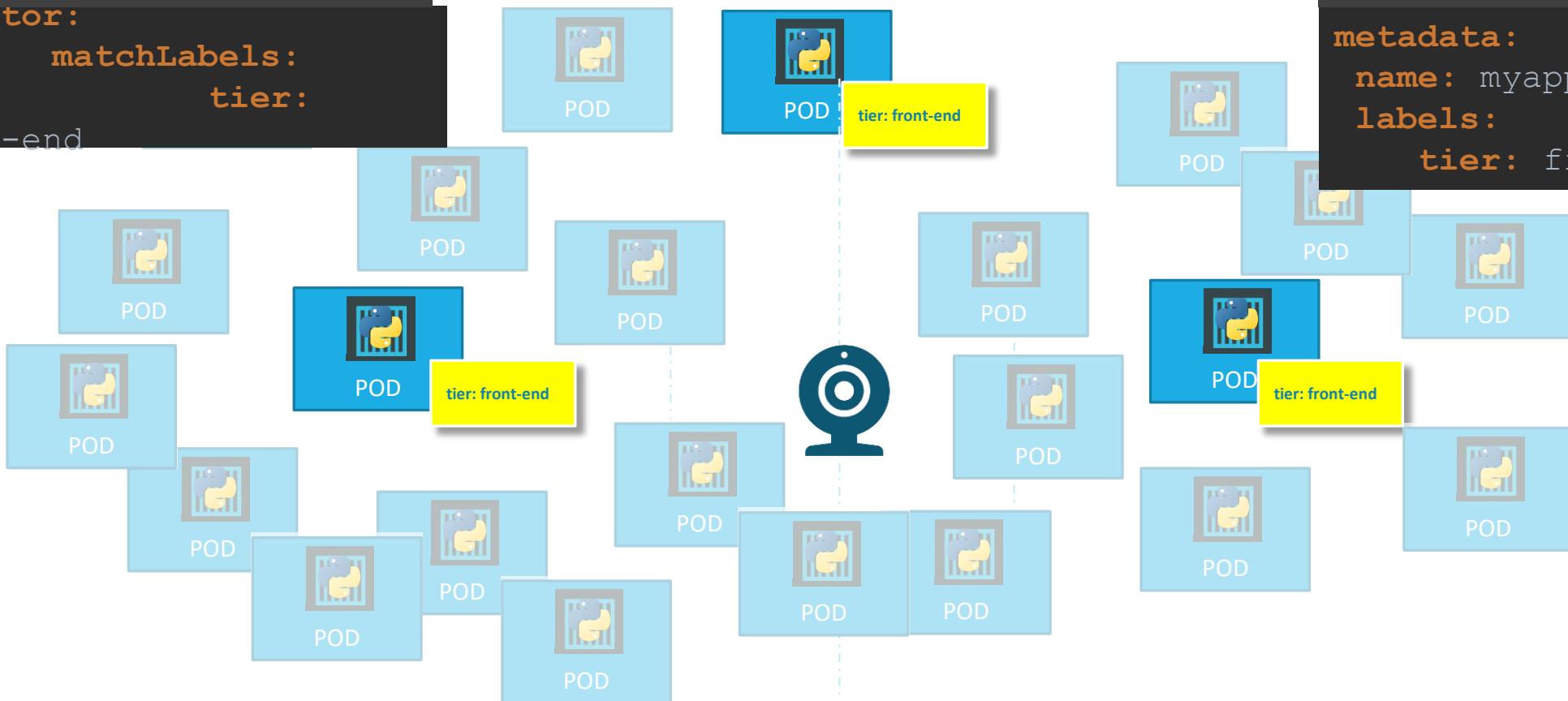
NAME	DESIRED	CURRENT	READY	AGE
myapp-replicaset	3	3	3	19s

```
> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
myapp-replicaset-9dd19	1/1	Running	0	45s
myapp-replicaset-9jtpx	1/1	Running	0	45s
myapp-replicaset-hq84m	1/1	Running	0	45s

Labels and Selectors

```
replicaset-definition.yml  
selector:  
  matchLabels:  
    tier: front-end
```



```
pod-definition.yml  
metadata:  
  name: myapp-pod  
labels:  
  tier: front-end
```

replicaset-definition.yml

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: myapp-replicaset
  labels:
    app: myapp
    type: front-end
spec:
  template:
    metadata:
      name: myapp-pod
      labels:
        app: myapp
        type: front-end
    spec:
      containers:
        - name: nginx-container
          image: nginx
  replicas: 3
  selector:
    matchLabels:
      type: front-end
```



Scale

```
> kubectl replace -f replicaset-definition.yml
```

```
> kubectl scale --replicas=6 -f replicaset-definition.yml
```

```
> kubectl scale --replicas=6 replicaset myapp-replicaset
```



```
replicaset-definition.yml
```

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: myapp-replicaset
  labels:
    app: myapp
    type: front-end
spec:
  template:
    metadata:
      name: myapp-pod
      labels:
        app: myapp
        type: front-end
    spec:
      containers:
      - name: nginx-container
        image: nginx
  replicas: 6
  selector:
    matchLabels:
      type: front-end
```

commands

```
> kubectl create -f replicaset-definition.yml
```

```
> kubectl get replicaset
```

```
> kubectl delete replicaset myapp-replicaset
```

*Also deletes all underlying PODs

```
> kubectl replace -f replicaset-definition.yml
```

```
> kubectl scale -replicas=6 -f replicaset-definition.yml
```

Demo

ReplicaSet

ReplicaSet as an Horizontal Pod Autoscaler Target

<https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/#replicaset-as-an-horizontal-pod-autoscaler-target>



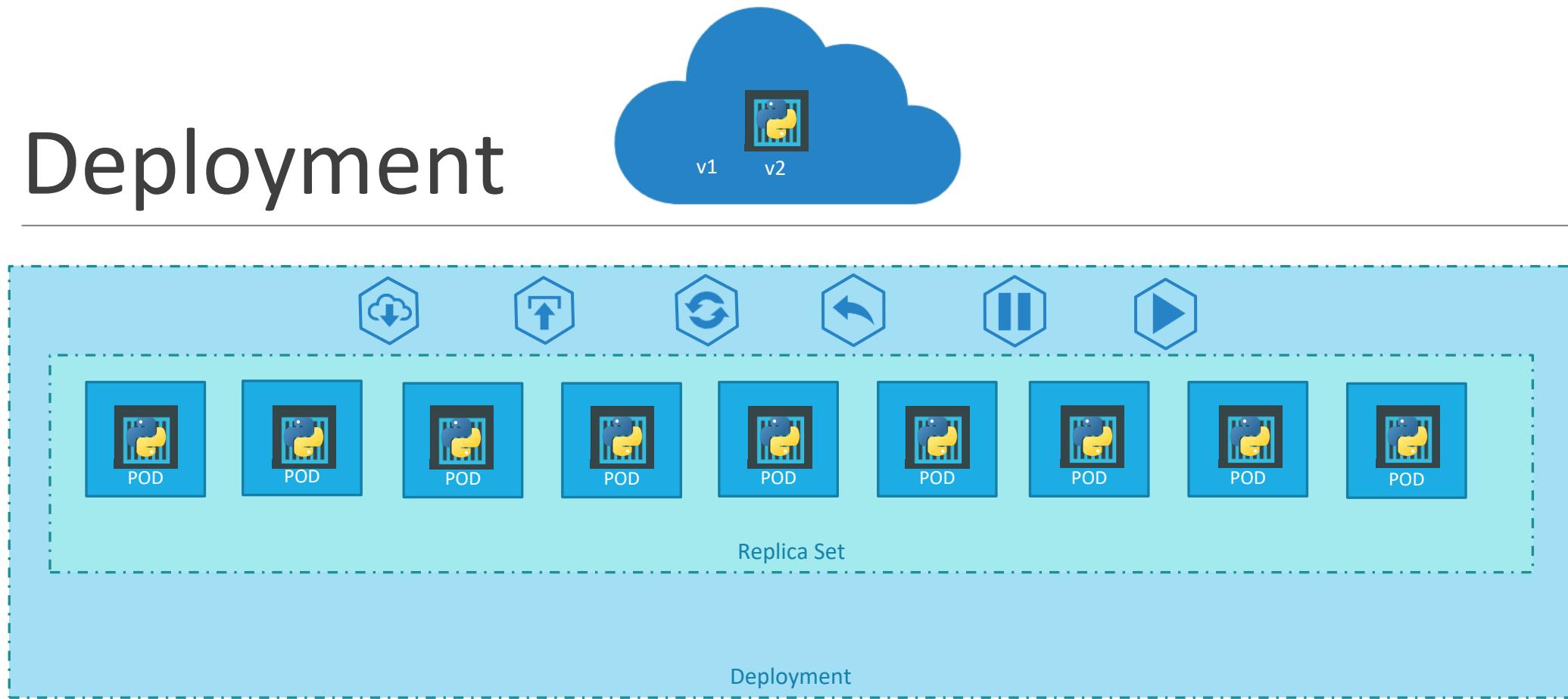
KodeKloud

Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

Deployment

mumshad mannambeth

Deployment



Definition

```
> kubectl create -f deployment-definition.yml  
deployment "myapp-deployment" created
```

```
> kubectl get deployments
```

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
myapp-deployment	3	3	3	3	21s

```
> kubectl get replicaset
```

NAME	DESIRED	CURRENT	READY	AGE
myapp-deployment-6795844b58	3	3	3	2m

```
> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
myapp-deployment-6795844b58-5rbj1	1/1	Running	0	2m
myapp-deployment-6795844b58-h4w55	1/1	Running	0	2m
myapp-deployment-6795844b58-1fjhv	1/1	Running	0	2m

```
deployment-definition.yml
```

```
apiVersion: apps/v1  
kind: ReplicaSet  
metadata:  
  name: myapp-deployment  
  labels:  
    app: myapp  
    type: front-end  
spec:  
  template:  
    metadata:  
      name: myapp-pod  
      labels:  
        app: myapp  
        type: front-end  
    spec:  
      containers:  
      - name: nginx-container  
        image: nginx  
    replicas: 3  
  selector:  
    matchLabels:  
      type: front-end
```

commands

```
> kubectl get all
```

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deploy/myapp-deployment	3	3	3	3	9h
<hr/>					
NAME	DESIRED	CURRENT	READY	AGE	
rs/myapp-deployment-6795844b58	3	3	3	9h	
<hr/>					
NAME	READY	STATUS	RESTARTS	AGE	
po/myapp-deployment-6795844b58-5rbjl	1/1	Running	0	9h	
po/myapp-deployment-6795844b58-h4w55	1/1	Running	0	9h	
po/myapp-deployment-6795844b58-1fjhv	1/1	Running	0	9h	

Demo

Deployment



KodeKloud

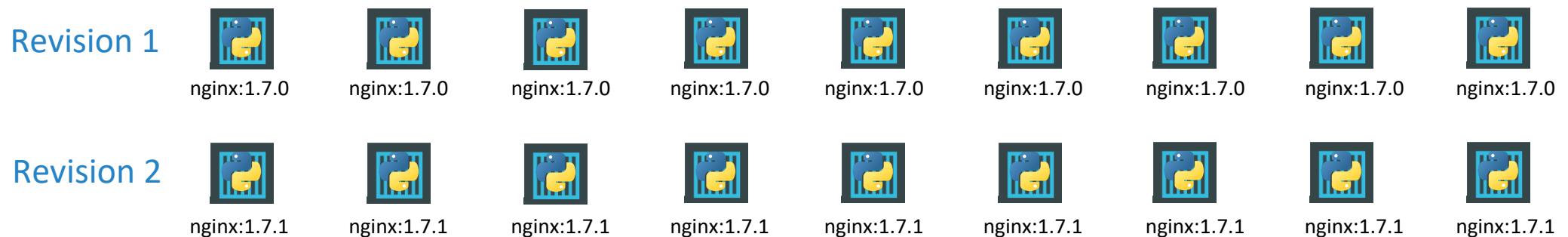
Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

Deployment

Updates and Rollback

mumshad mannambeth

Rollout and Versioning

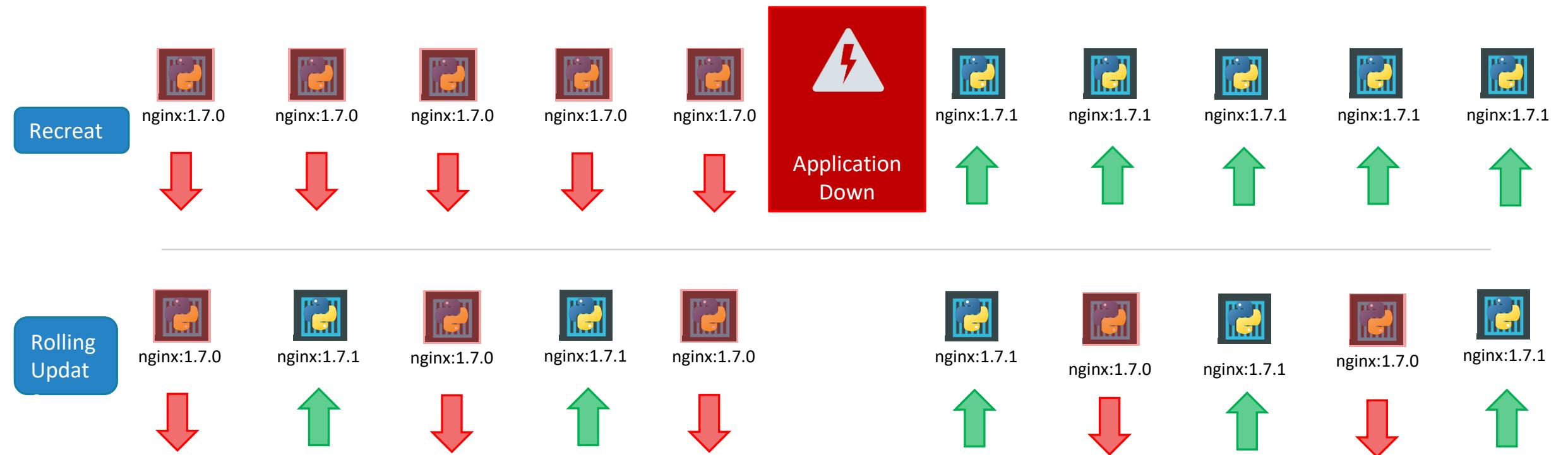


Rollout Command

```
> kubectl rollout status deployment/myapp-deployment
Waiting for rollout to finish: 0 of 10 updated replicas are available...
Waiting for rollout to finish: 1 of 10 updated replicas are available...
Waiting for rollout to finish: 2 of 10 updated replicas are available...
Waiting for rollout to finish: 3 of 10 updated replicas are available...
Waiting for rollout to finish: 4 of 10 updated replicas are available...
Waiting for rollout to finish: 5 of 10 updated replicas are available...
Waiting for rollout to finish: 6 of 10 updated replicas are available...
Waiting for rollout to finish: 7 of 10 updated replicas are available...
Waiting for rollout to finish: 8 of 10 updated replicas are available...
Waiting for rollout to finish: 9 of 10 updated replicas are available...
deployment "myapp-deployment" successfully rolled out
```

```
> kubectl rollout history deployment/myapp-deployment
deployments "myapp-deployment"
REVISION  CHANGE-CAUSE
1        <none>
2        kubectl apply --filename=deployment-definition.yml --record=true
```

Deployment Strategy



Kubectl apply

```
> kubectl apply -f deployment-definition.yml  
deployment "myapp-deployment" configured
```

```
> kubectl set image deployment/myapp-deployment \  
      nginx-container=nginx:1.9.1  
deployment "myapp-deployment" image is updated
```

```
deployment-definition.yml  
  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: myapp-deployment  
  labels:  
    app: myapp  
    type: front-end  
spec:  
  template:  
    metadata:  
      name: myapp-pod  
      labels:  
        app: myapp  
        type: front-end  
    spec:  
      containers:  
      - name: nginx-container  
        image: nginx:1.7.1  
      replicas: 3  
      selector:  
        matchLabels:  
          type: front-end
```

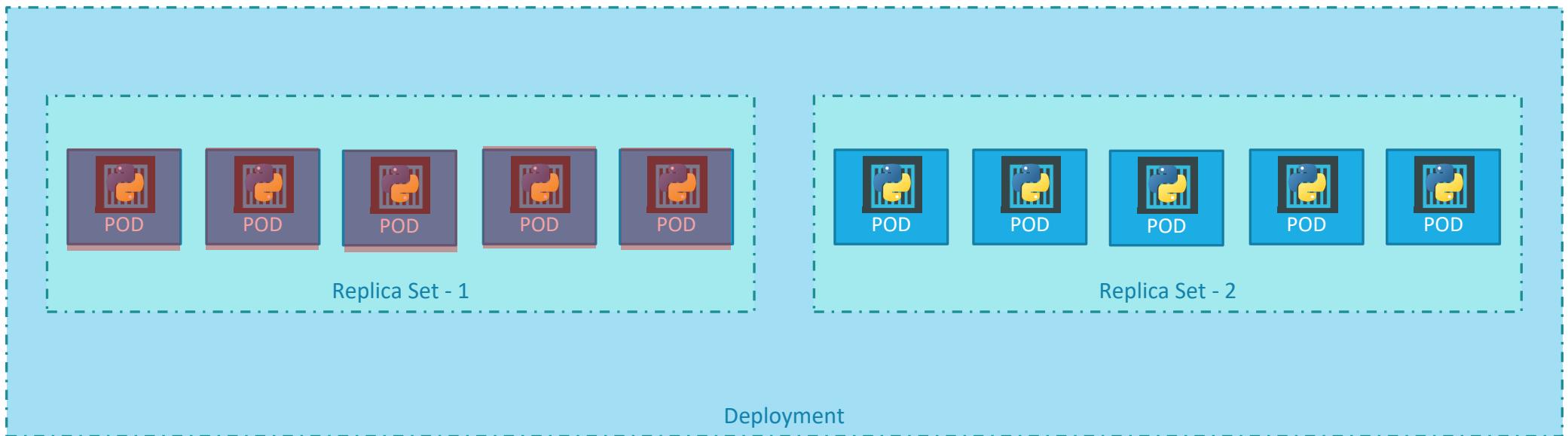
```
C:\Kubernetes>kubectl describe deployment myapp-deployment
Name:           myapp-deployment
Namespace:      default
CreationTimestamp: Sat, 03 Mar 2018 17:01:55 +0800
Labels:         app=myapp
                type=front-end
Annotations:   deployment.kubernetes.io/revision=2
                kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"apps/v1","kind":"Deployment","me...
s\\Google...
                kubernetes.io/change-cause=kubectl apply --filename=d:\Mumshad Files\Google Drive\Udemy\Kubernetes...
Selector:      type=front-end
Replicas:      5 desired | 5 updated | 5 total | 5 available | 0 unavailable
StrategyType:  Recreate
MinReadySeconds: 0
Pod Template:
  Labels:  app=myapp
            type=front-end
  Containers:
    nginx-container:
      Image:  nginx:1.7.1
      Port:   <none>
      Environment: <none>
      Mounts:  <none>
      Volumes: <none>
  Conditions:
    Type     Status  Reason
    ----  -----
    Available  True    MinimumReplicasAvailable
    Progressing True   NewReplicaSetAvailable
OldReplicaSets: <none>
NewReplicaSet:  myapp-deployment-54c7d6ccc (5/5 replicas created)
Events:
  Type    Reason     Age   From               Message
  ----  -----  -----
  Normal  ScalingReplicaSet 11m  deployment-controller  Scaled up replica set myapp-deployment-6795844b58 to 5
  Normal  ScalingReplicaSet 1m   deployment-controller  Scaled down replica set myapp-deployment-6795844b58 to 0
  Normal  ScalingReplicaSet 56s  deployment-controller  Scaled up replica set myapp-deployment-54c7d6ccc to 5
```

Recreate

```
C:\Kubernetes>kubectl describe deployment myapp-deployment
Name:           myapp-deployment
Namespace:      default
CreationTimestamp: Sat, 03 Mar 2018 17:16:53 +0800
Labels:         app=myapp
                type=front-end
Annotations:   deployment.kubernetes.io/revision=2
                kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"apps/v1","kind":"Deployment","me...
Files\\Google...
                kubernetes.io/change-cause=kubectl apply --filename=d:\Mumshad Files\Google Drive\Udemy\Kubernetes...
Selector:      type=front-end
Replicas:      5 desired | 5 updated | 6 total | 4 available | 2 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=myapp
            type=front-end
  Containers:
    nginx-container:
      Image:  nginx
      Port:   <none>
      Environment: <none>
      Mounts:  <none>
      Volumes: <none>
  Conditions:
    Type     Status  Reason
    ----  -----
    Available  True    MinimumReplicasAvailable
    Progressing True   ReplicaSetUpdated
OldReplicaSets: myapp-deployment-67c749c58c (1/1 replicas created)
NewReplicaSet:  myapp-deployment-7d57dbdb8d (5/5 replicas created)
Events:
  Type    Reason     Age   From               Message
  ----  -----  -----
  Normal  ScalingReplicaSet 1m  deployment-controller  Scaled up replica set myapp-deployment-67c749c58c to 5
  Normal  ScalingReplicaSet 1s  deployment-controller  Scaled up replica set myapp-deployment-7d57dbdb8d to 2
  Normal  ScalingReplicaSet 1s  deployment-controller  Scaled down replica set myapp-deployment-67c749c58c to 4
  Normal  ScalingReplicaSet 1s  deployment-controller  Scaled up replica set myapp-deployment-7d57dbdb8d to 3
  Normal  ScalingReplicaSet 0s  deployment-controller  Scaled down replica set myapp-deployment-67c749c58c to 3
  Normal  ScalingReplicaSet 0s  deployment-controller  Scaled up replica set myapp-deployment-7d57dbdb8d to 4
  Normal  ScalingReplicaSet 0s  deployment-controller  Scaled down replica set myapp-deployment-67c749c58c to 2
  Normal  ScalingReplicaSet 0s  deployment-controller  Scaled up replica set myapp-deployment-7d57dbdb8d to 5
  Normal  ScalingReplicaSet 0s  deployment-controller  Scaled down replica set myapp-deployment-67c749c58c to 1
```

RollingUpdate

Upgrades



```
> kubectl get replicaset
```

NAME	DESIRED	CURRENT	READY	AGE
myapp-deployment-67c749c58c	0	0	0	22m
myapp-deployment-7d57dbdb8d	5	5	5	20m

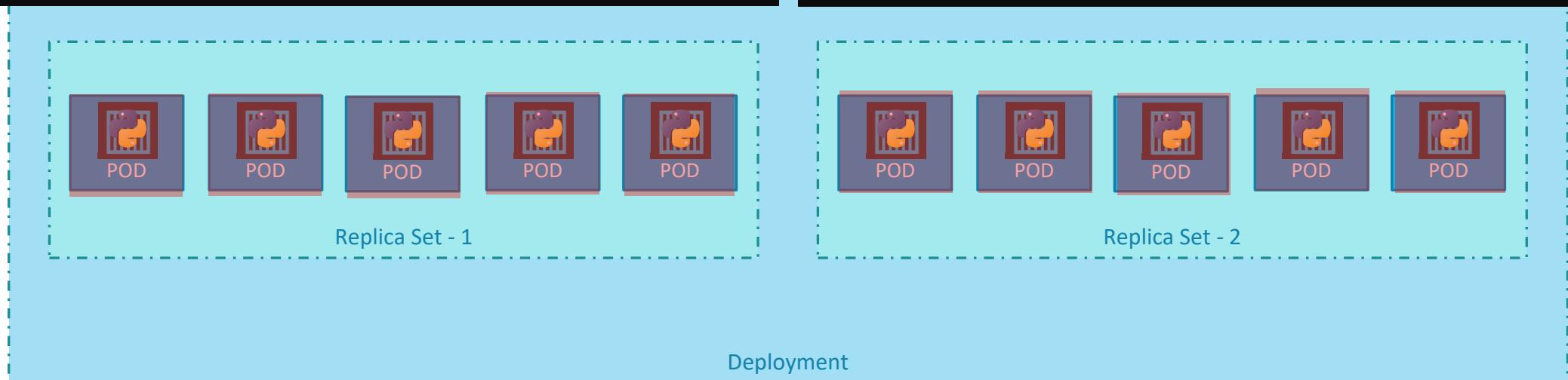
Rollback

```
> kubectl get replicsets
```

NAME	DESIRED	CURRENT	READY	AGE
myapp-deployment-67c749c58c	0	0	0	22m
myapp-deployment-7d57dbdb8d	5	5	5	20m

```
> kubectl get replicsets
```

NAME	DESIRED	CURRENT	READY	AGE
myapp-deployment-67c749c58c	5	5	5	22m
myapp-deployment-7d57dbdb8d	0	0	0	20m



```
> kubectl rollout undo deployment/myapp-deployment  
deployment “myapp-deployment” rolled back
```

kubectl run

```
> kubectl run nginx --image=nginx
```

```
deployment "nginx" created
```

Summarize Commands

Create

```
> kubectl create -f deployment-definition.yml
```

Get

```
> kubectl get deployments
```

Update

```
> kubectl apply -f deployment-definition.yml
```

```
> kubectl set image deployment/myapp-deployment nginx=nginx:1.9.1
```

Status

```
> kubectl rollout status deployment/myapp-deployment
```

```
> kubectl rollout history deployment/myapp-deployment
```

Rollback

```
> kubectl rollout undo deployment/myapp-deployment
```

Demo

Deployment



KodeKloud

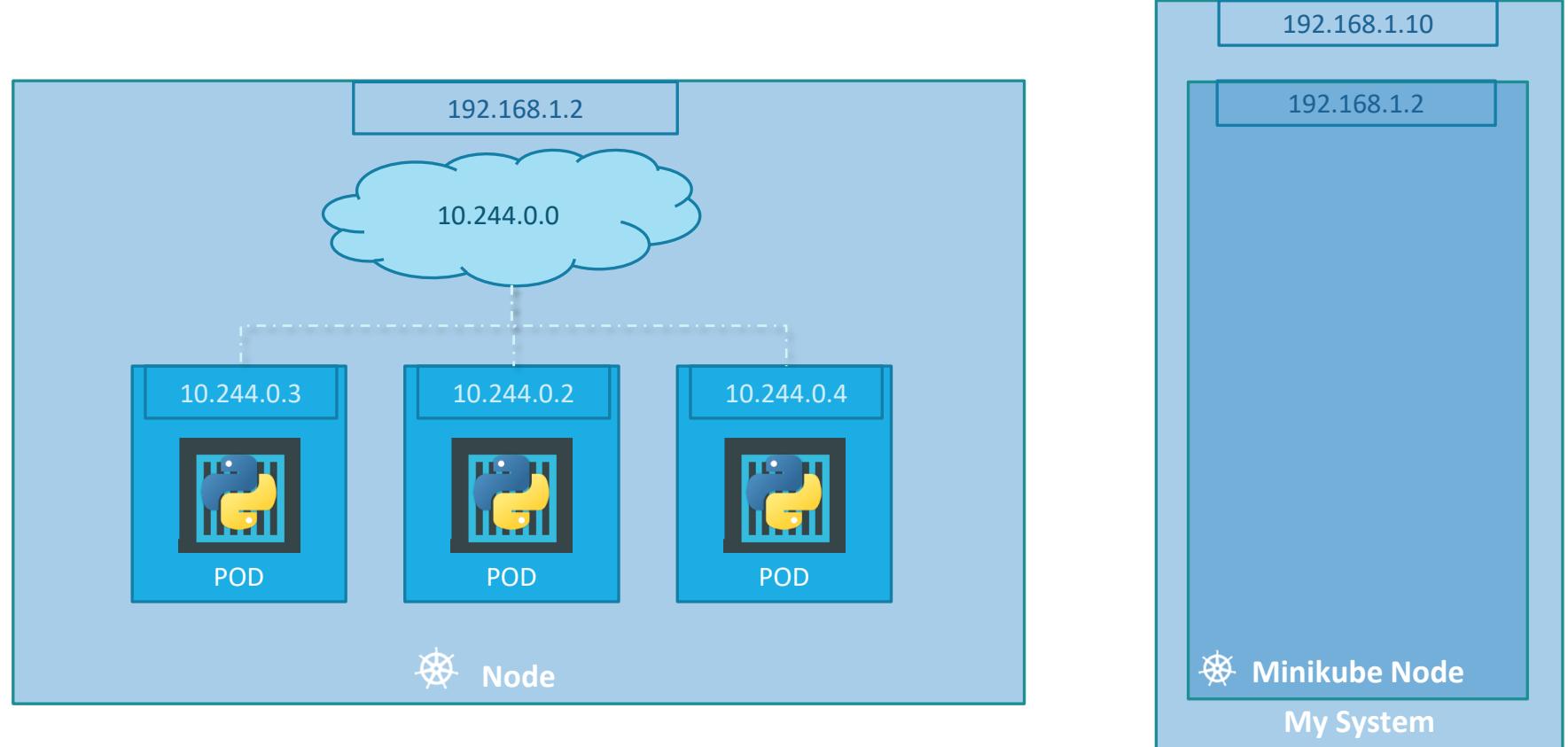
Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

Networking 101

mumshad mannambeth

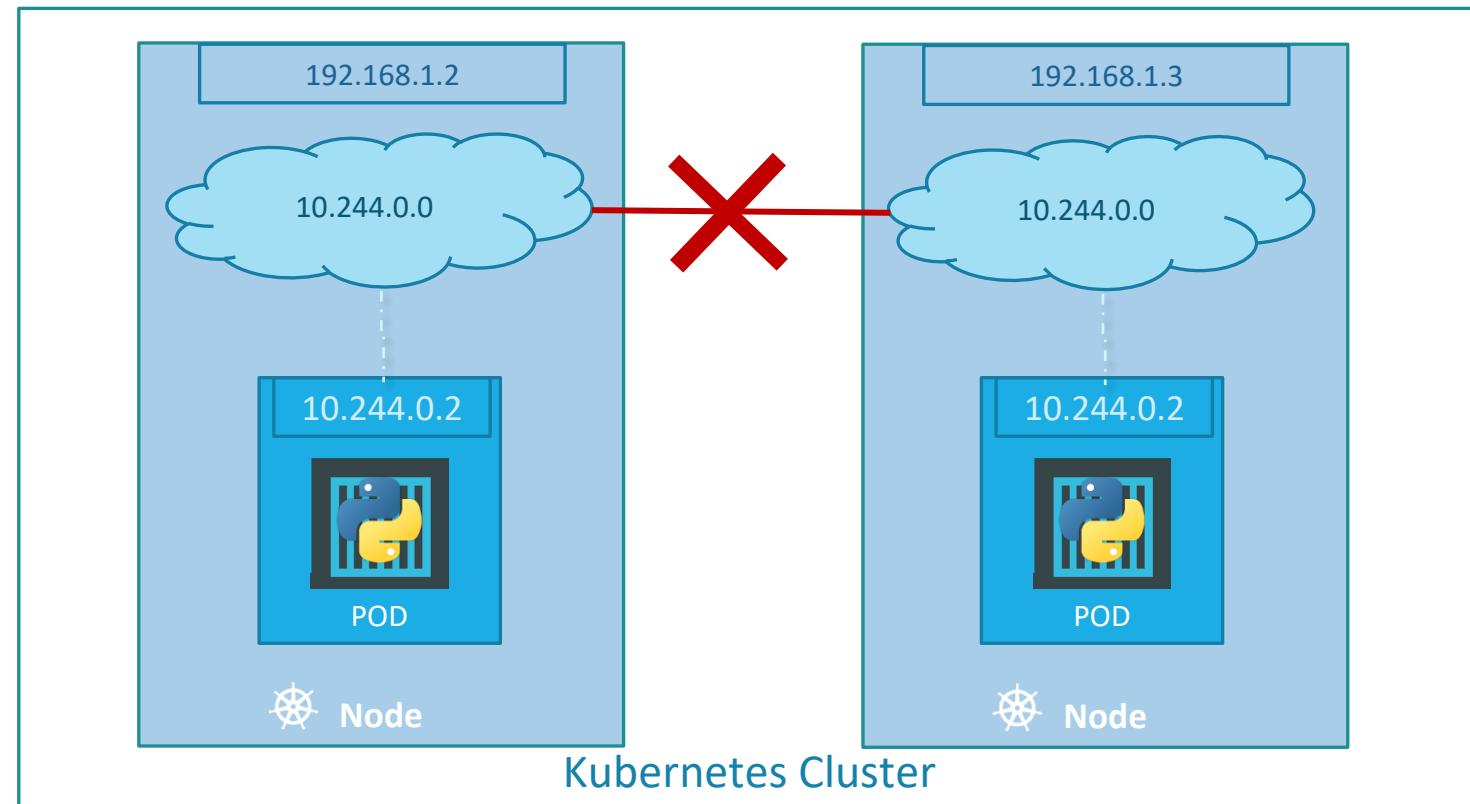
Kubernetes Networking - 101

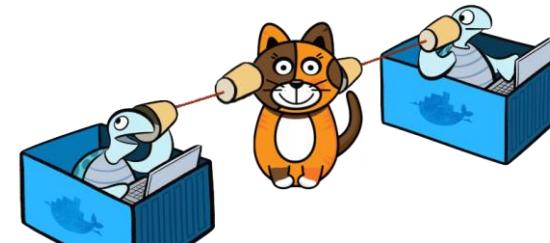
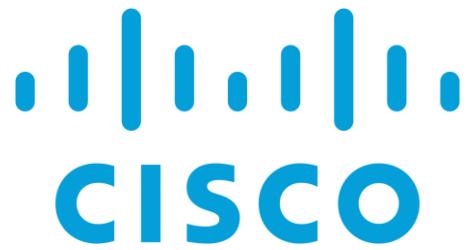
- IP Address is assigned to a POD



Cluster Networking

- All containers/PODs can communicate to one another without NAT
- All nodes can communicate with all containers and vice-versa without NAT





Cluster Networking Setup

(3/4) Installing a pod network

You **MUST** install a pod **network** add-on so that your pods can communicate with each other.

The **network** must be deployed before any applications. Also, kube-dns, an internal helper service, will not start up before a **network** is installed. kubeadm only supports Container Network Interface (CNI) based **networks** (and does not support kubenet).

Several projects provide Kubernetes pod **networks** using CNI, some of which also support **Network Policy**. See the [add-ons page](#) for a complete list of available **network** add-ons. IPv6 support was added in [CNI v0.6.0](#). [CNI bridge](#) and [local-ipam](#) are the only supported IPv6 **network** plugins in 1.9.

Note: kubeadm sets up a more secure cluster by default and enforces use of [RBAC](#). Please make sure that the **network** manifest of choice supports RBAC.

You can install a pod **network** add-on with the following command:

```
kubectl apply -f <add-on.yaml>
```

NOTE: You can install **only one** pod **network** per cluster.



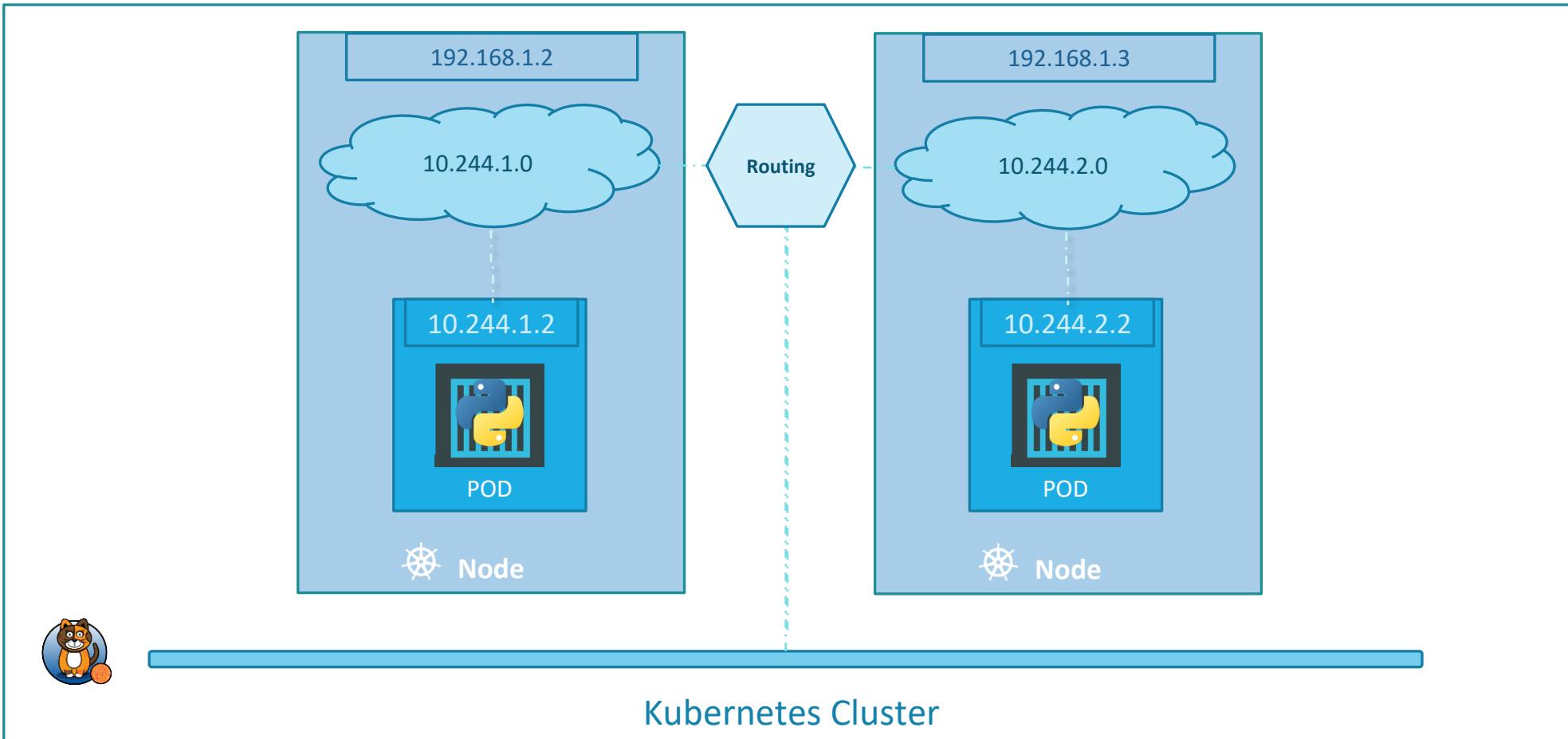
Refer to the Calico documentation for a [kubeadm quickstart](#), a [kubeadm installation guide](#), and other resources.

Note:

- In order for Network Policy to work correctly, you need to pass `--pod-network-cidr=192.168.0.0/16` to `kubeadm init`.
- Calico works on `amd64` only.

```
kubectl apply -f https://docs.projectcalico.org/v3.0/getting-started/kubernetes/installation/hosted/kubeadm/1.7/calico.yaml
```

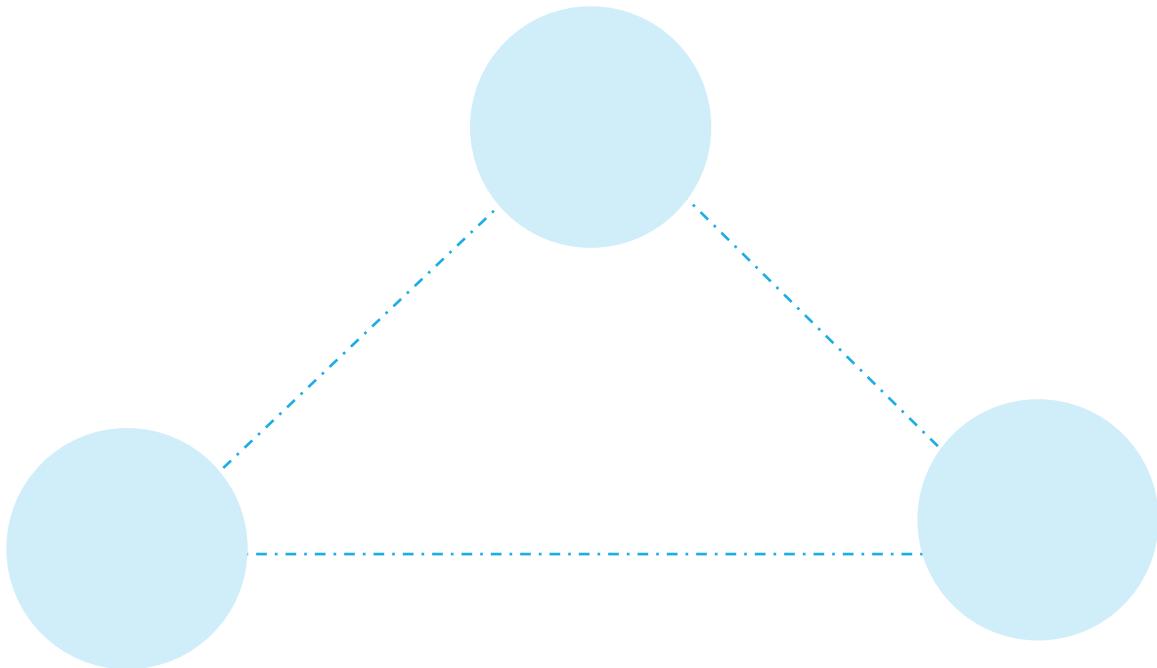
Cluster Networking



Demo

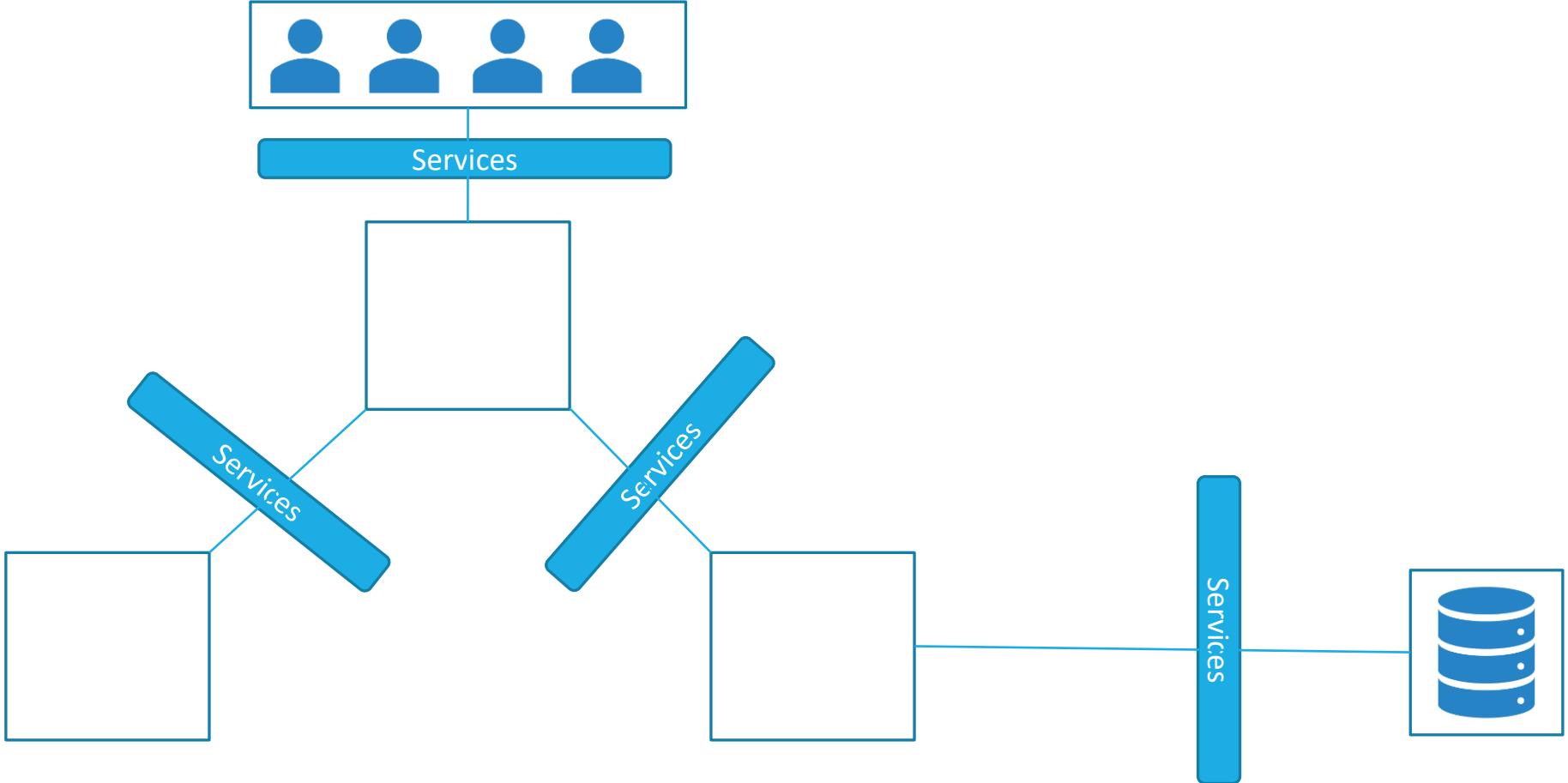
Networking

Services

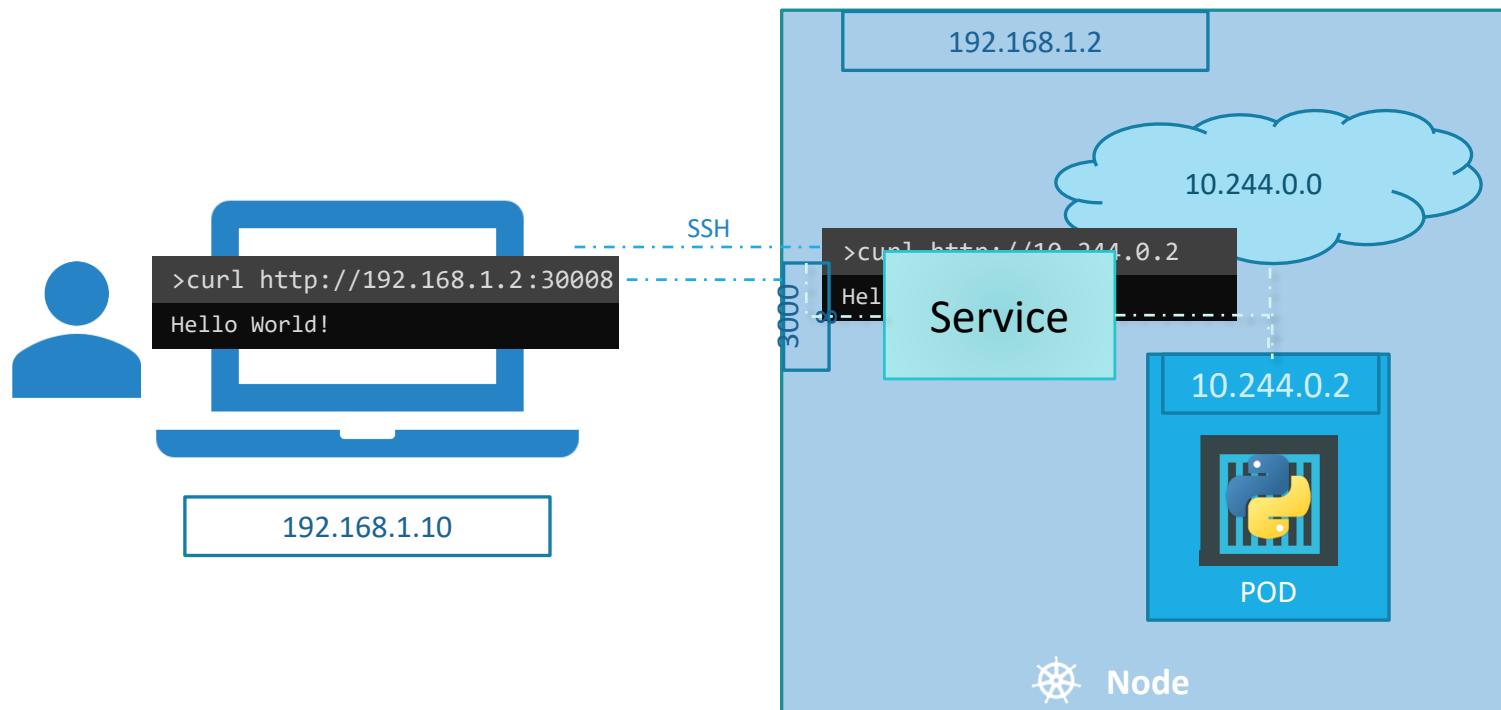


mumshad mannambeth

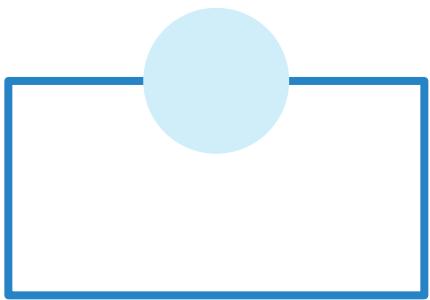
Services



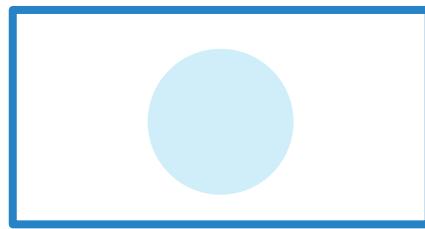
Service



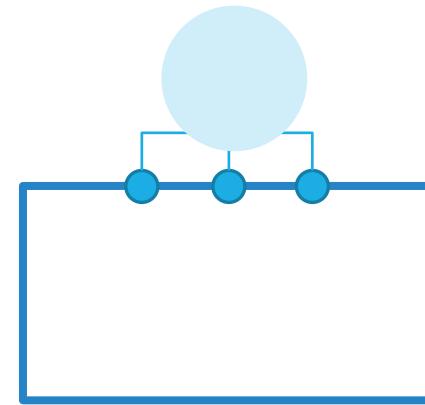
Services Types



NodePort



ClusterIP



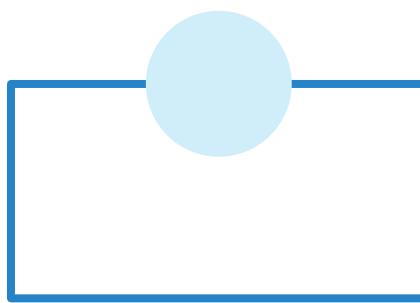
LoadBalancer



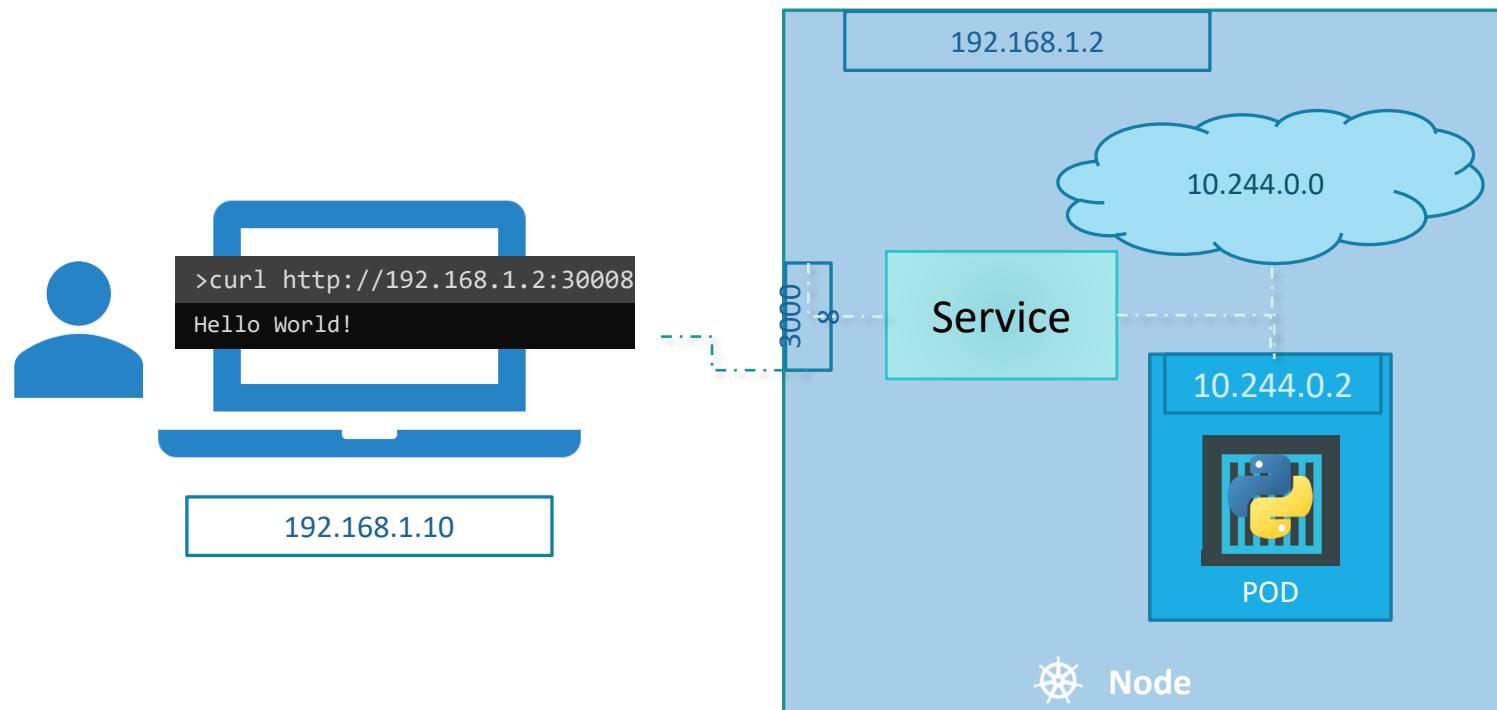
KodeKloud

Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

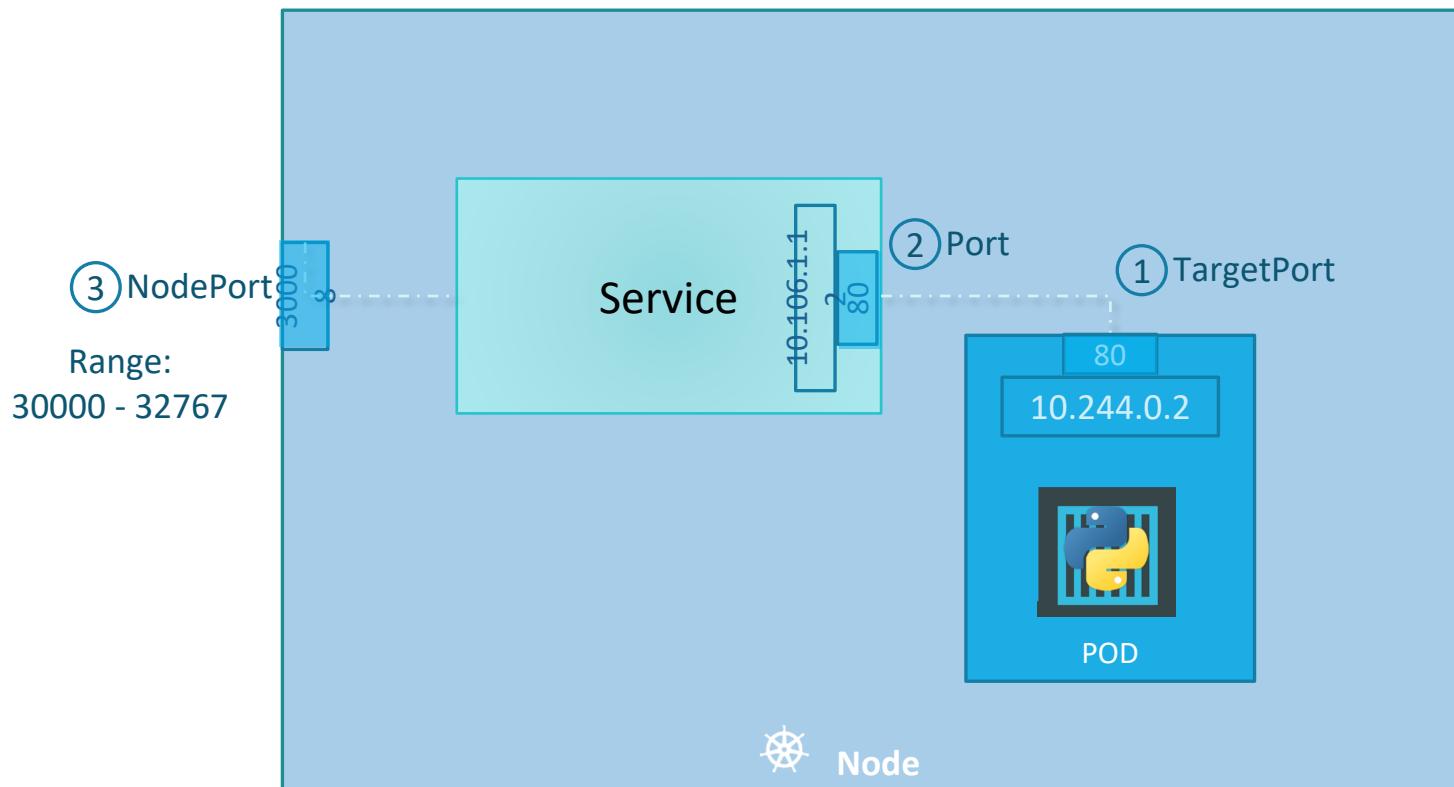
NodePort



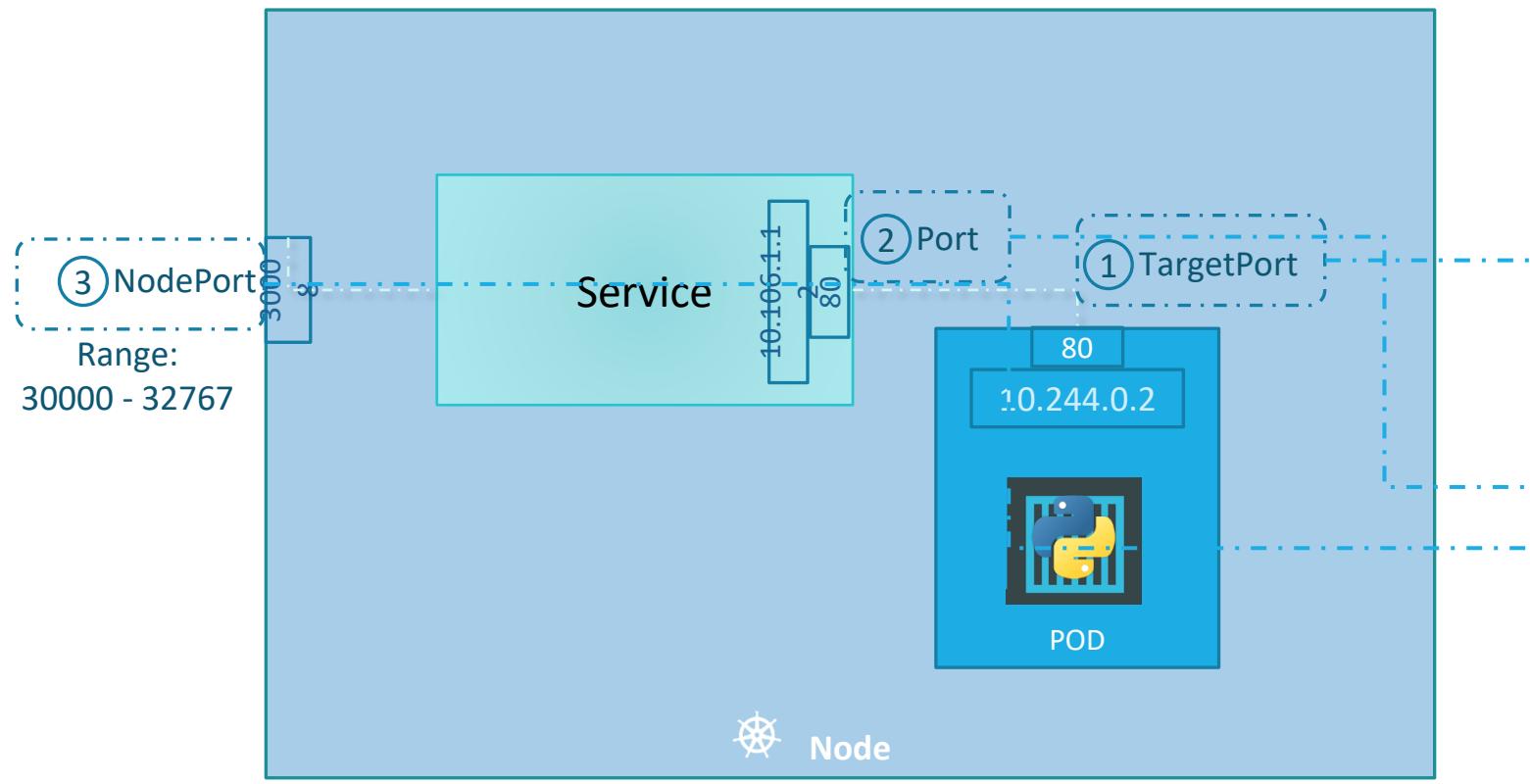
Service - NodePort



Service - NodePort



Service - NodePort



`service-definition.yml`

```
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: NodePort
  ports:
    - targetPort: 80
      *port: 80
      nodePort: 30008
```

Service - NodePort

service-definition.yml

```
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:
```

pod-definition.yml

```
> kubectl create -f service-definition.yml
service "myapp-service" created
```

```
> kubectl get services
```

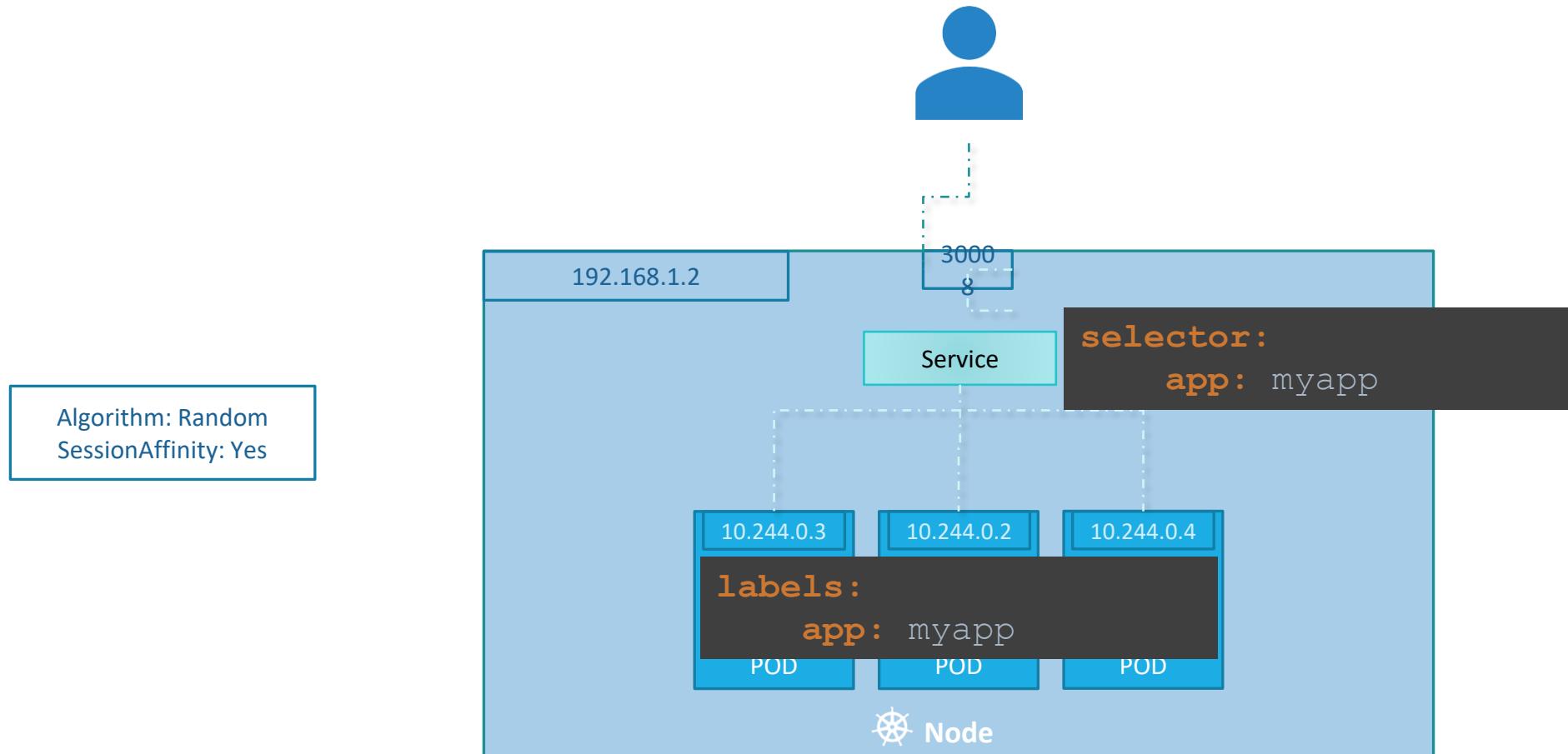
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	16d
myapp-service	NodePort	10.106.127.123	<none>	80:30008/TCP	5m

app: myapp

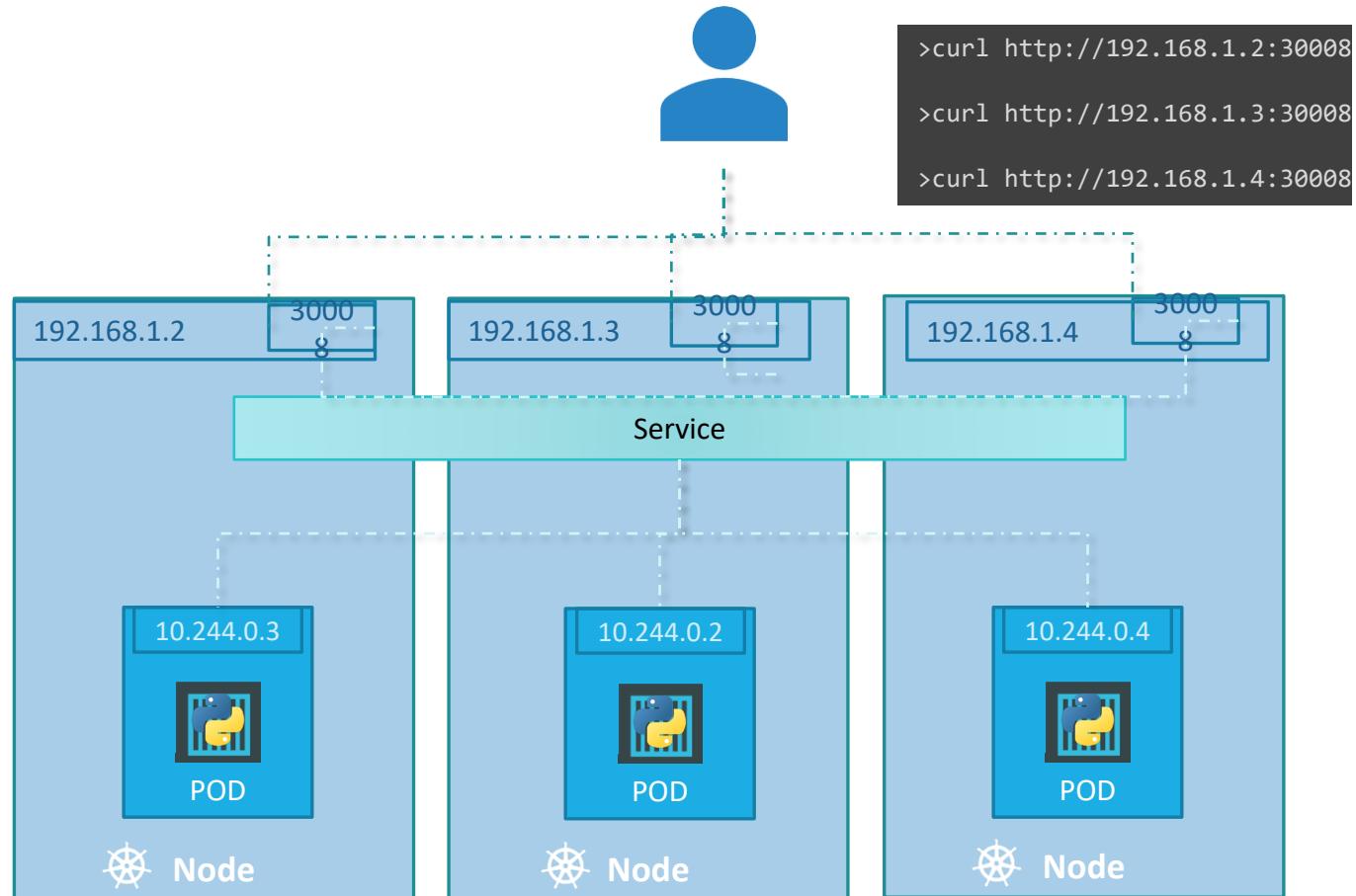
```
> curl http://192.168.1.2:30008
```

```
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
```

Service - NodePort



Service - NodePort



Demo

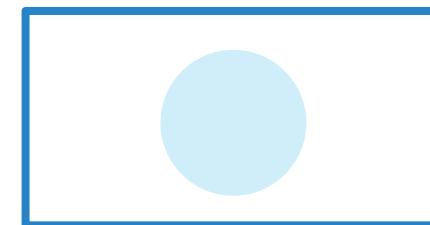
Service - NodePort



KodeKloud

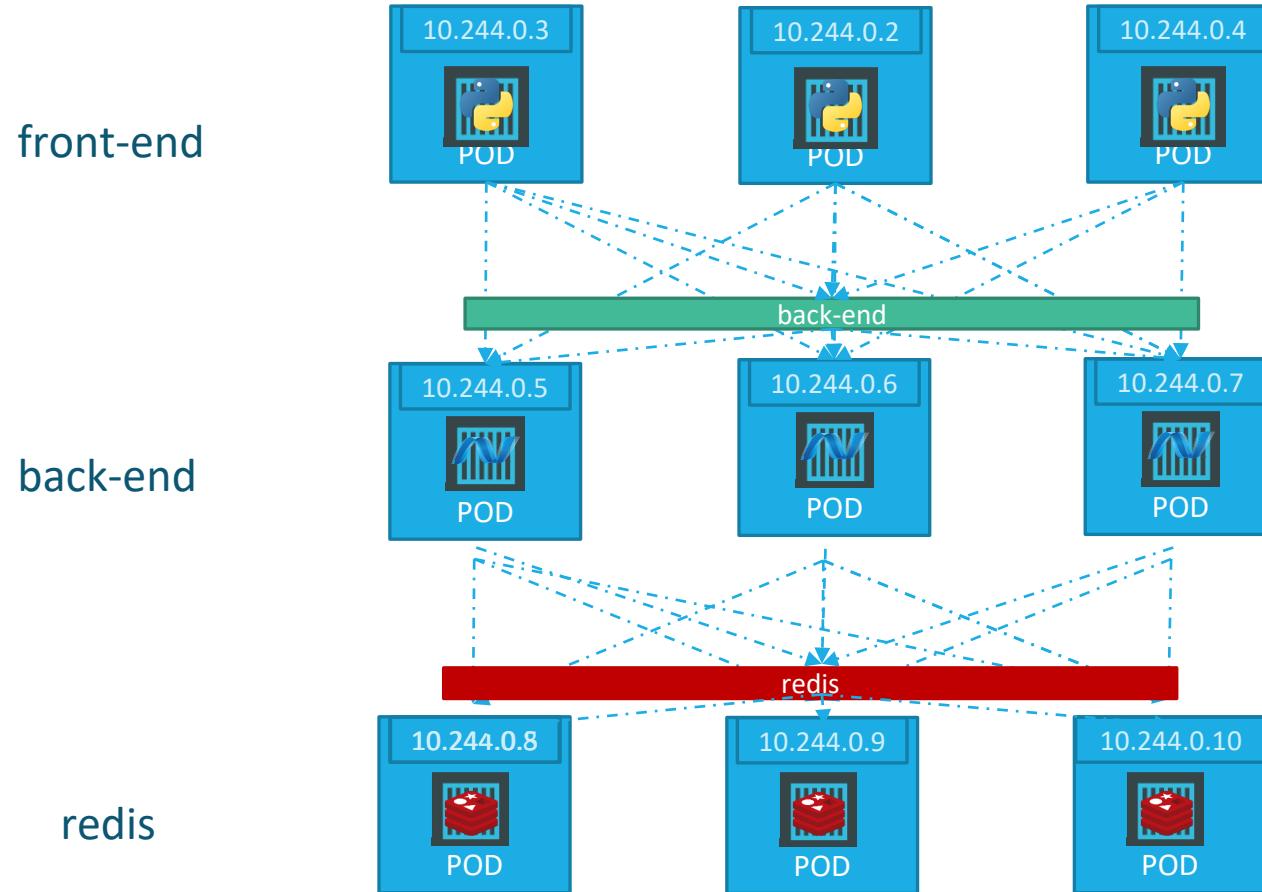
Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

ClusterIP



mumshad mannambeth

ClusterIP



service-definition.yml

```
apiVersion: v1
kind: Service
metadata:
  name: back-end
spec:
  type: ClusterIP
  ports:
    - targetPort: 80
      port: 80
  selector:
```

pod-definition.yml

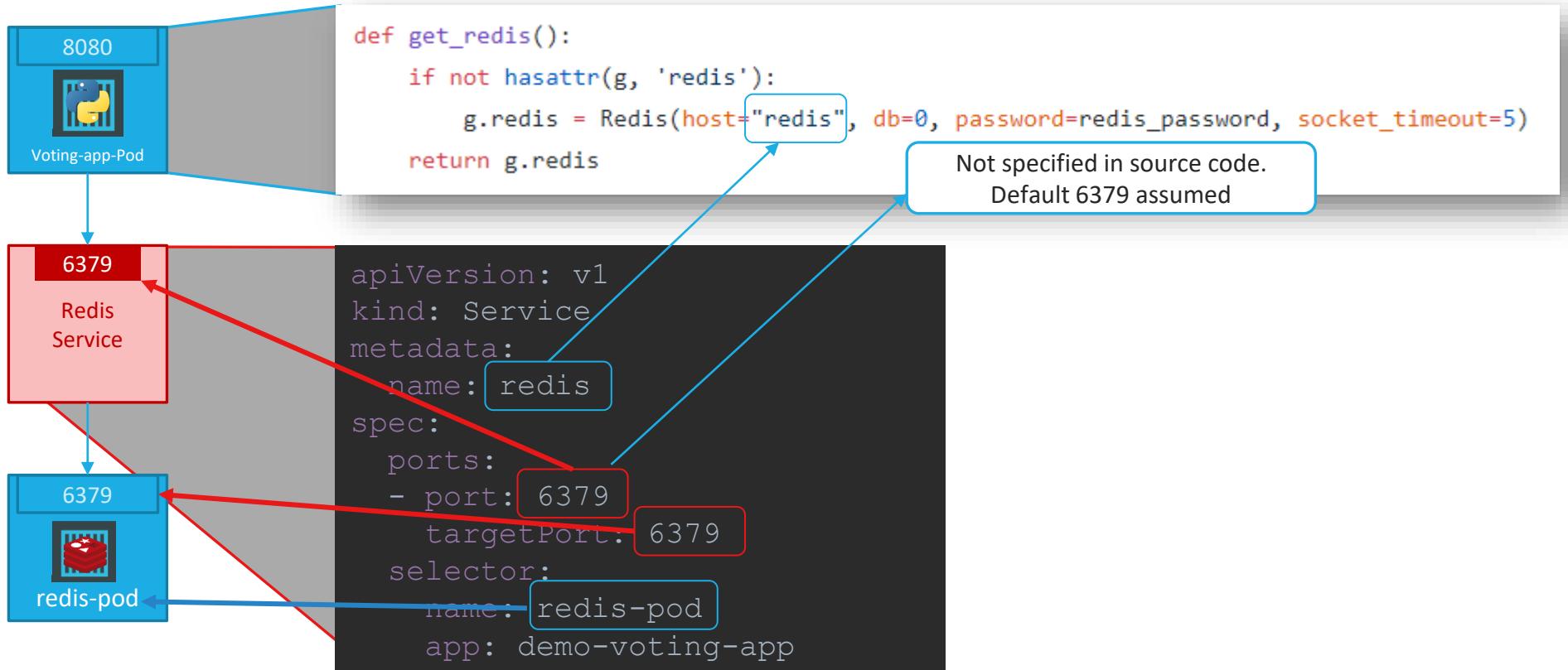
```
> kubectl create -f service-definition.yml
service "back-end" created
```

```
> kubectl get services
```

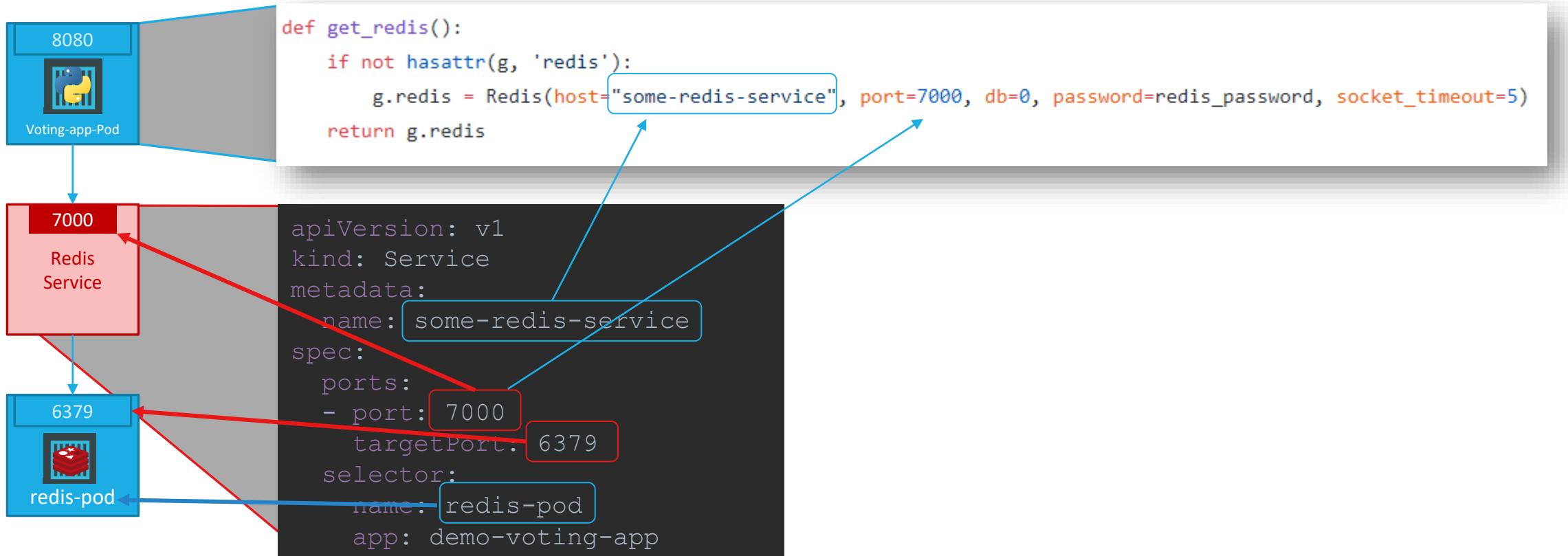
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	16d
back-end	ClusterIP	10.106.127.123	<none>	80/TCP	2m

```
app: myapp
type: back-end
spec:
  containers:
    - name: nginx-container
      image: nginx
```

Service



Service



Demo

Service - NodePort

References

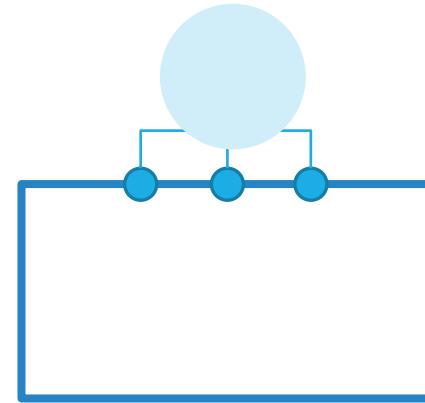
<https://kubernetes.io/docs/concepts/services-networking/dns-pod-service/>



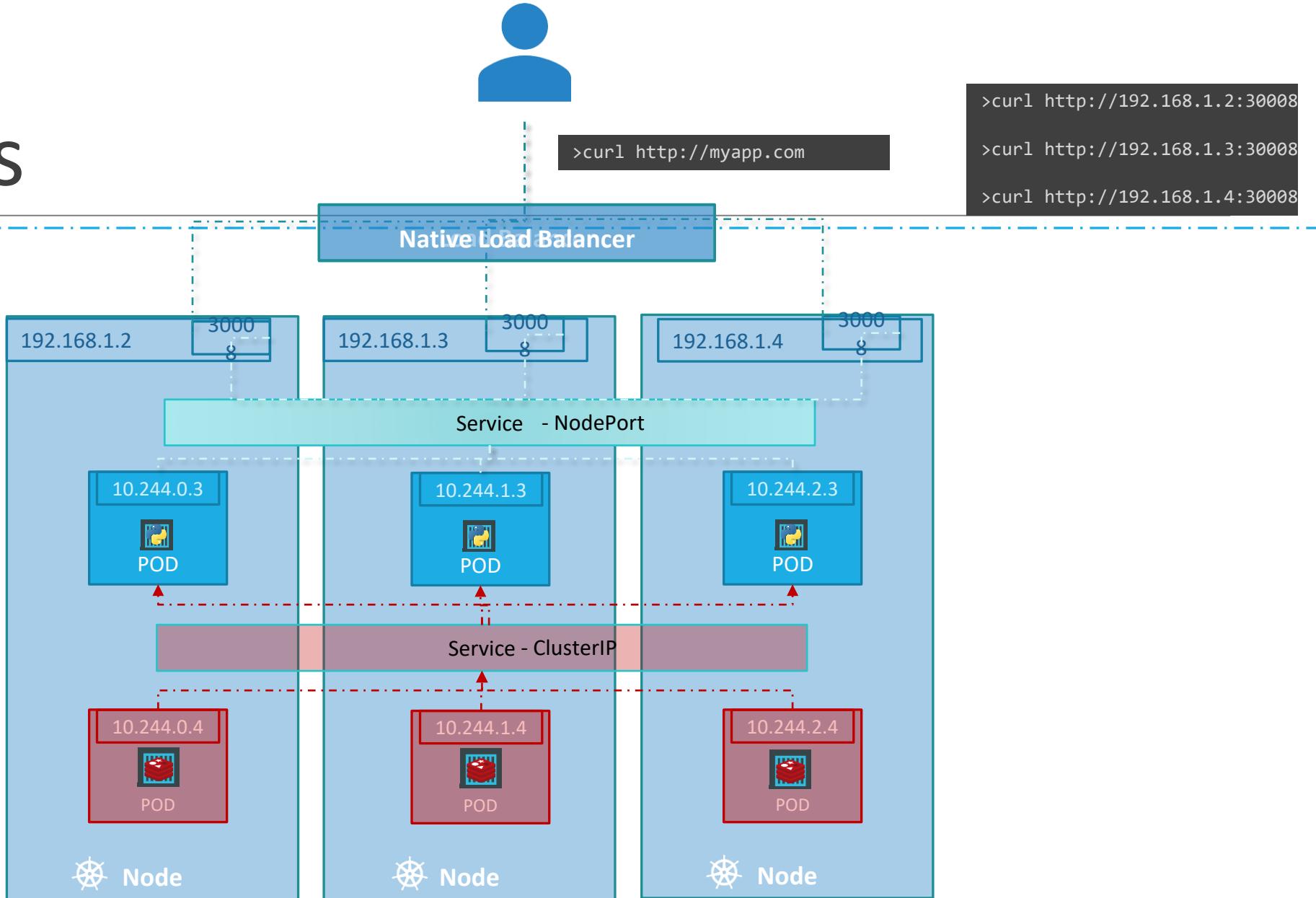
KodeKloud

Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

Service - LoadBalancer



Services



service-definition.yml

```
apiVersion: v1
kind: Service
metadata:
  name: front-end
spec:
  type: NodeBalancer
  ports:
    - targetPort: 80
      port: 80

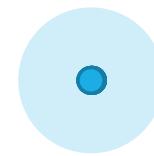
  selector:
    app: myapp
    type: front-end
```

```
> kubectl create -f service-definition.yml
service "front-end" created
```

```
> kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	16d
front-end	LoadBalancer	10.106.127.123	<Pending>	80/TCP	2m

Microservices



mumshad mannambeth

Example voting app

app.py

```
app = Flask(__name__)

def get_redis():
    if not hasattr(g, 'redis'):
        g.redis = Redis(host="redis", db=0, socket_timeout=5)
    return g.redis
```

Steps:

1. Deploy PODs
2. Create Services (ClusterIP)
 1. redis
 2. db

program.cs Services (NodePort)

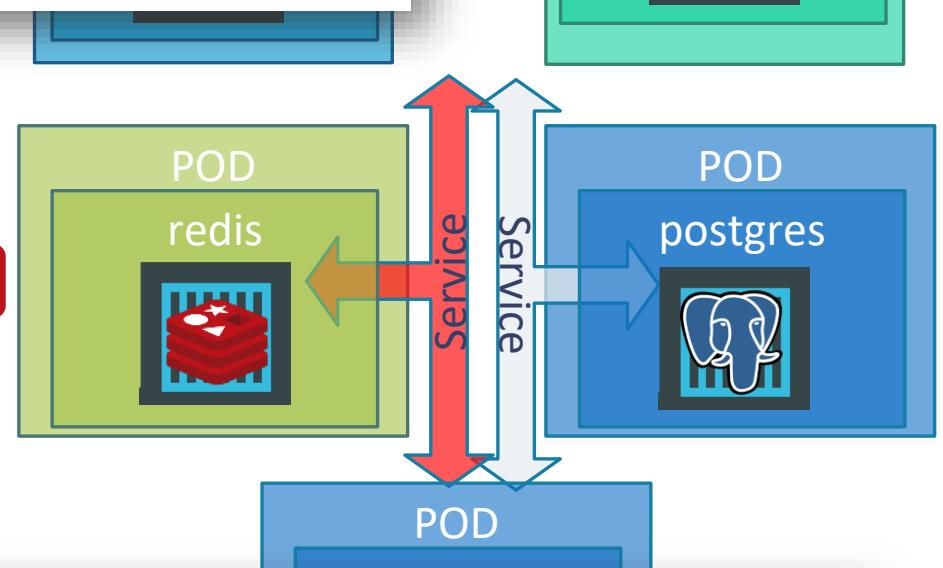
```
var pgsql = OpenDbConnection("Server=db;Username=postgres;Password=postgres;");
var redisConn = OpenRedisConnection("redis");
var redis = redisConn.GetDatabase();
```



server.js

```
var pool = new pg.Pool({
  connectionString: 'postgres://postgres:postgres@db/postgres'
});
```

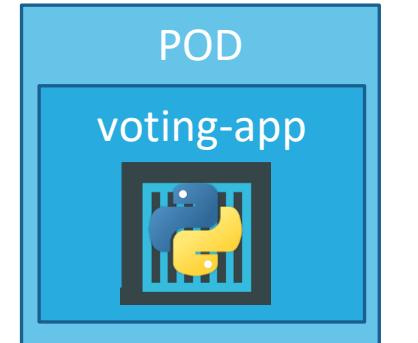
80



kodekloudhub / example-voting-app

forked from [dockersamples/example-voting-app](#)

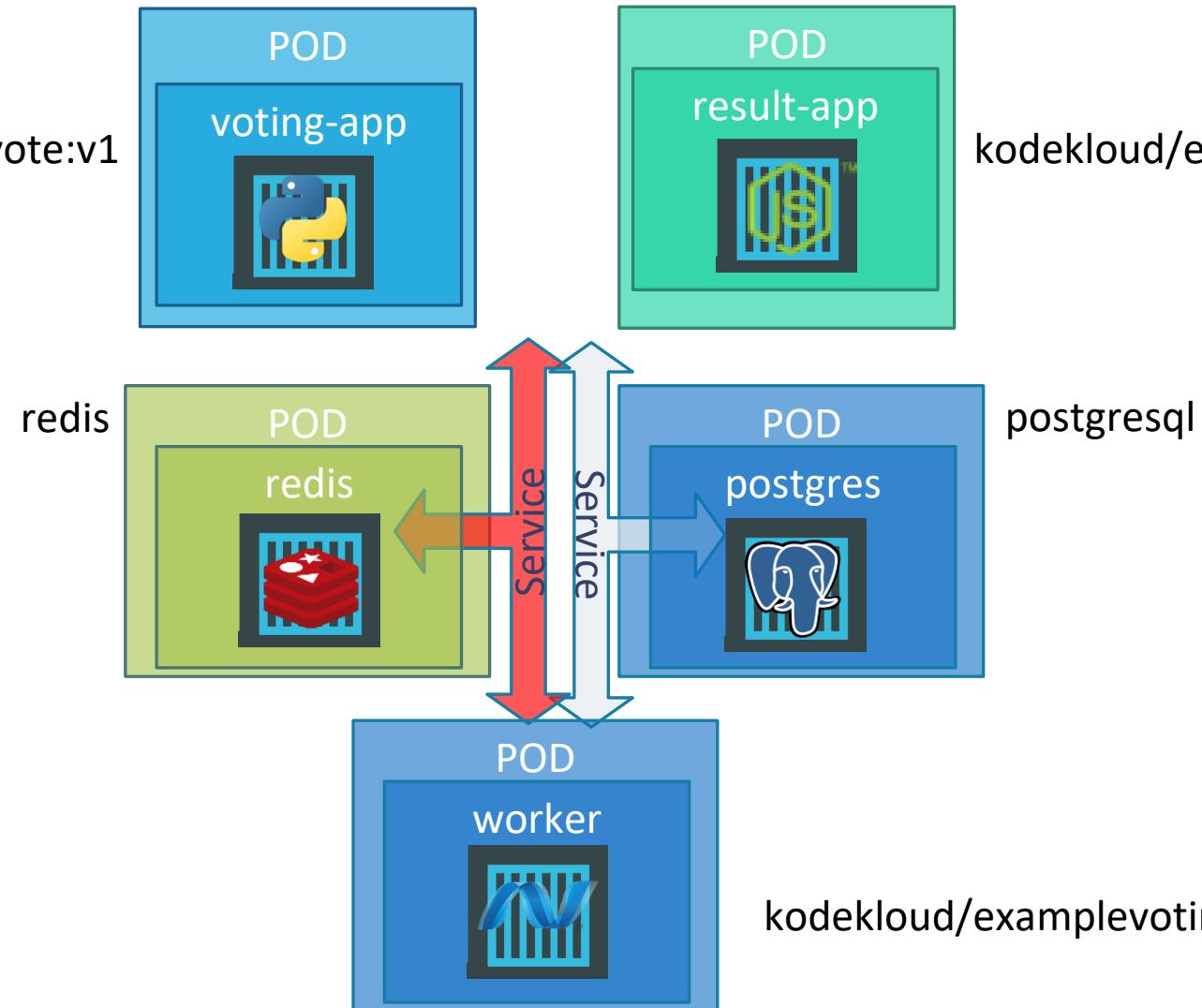
kodekloud/examplevotingapp_vote:v1



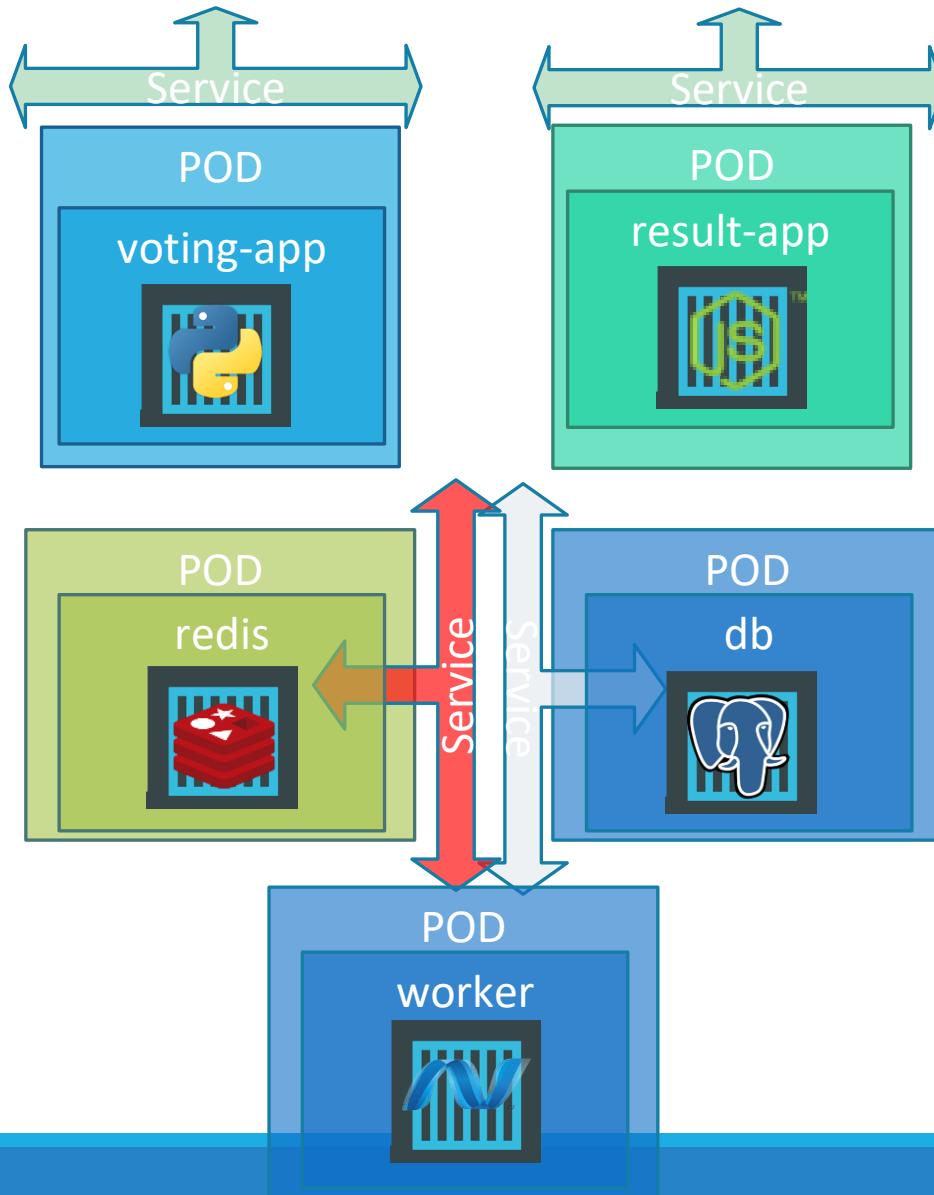
kodekloud/examplevotingapp_result:v1

Steps:

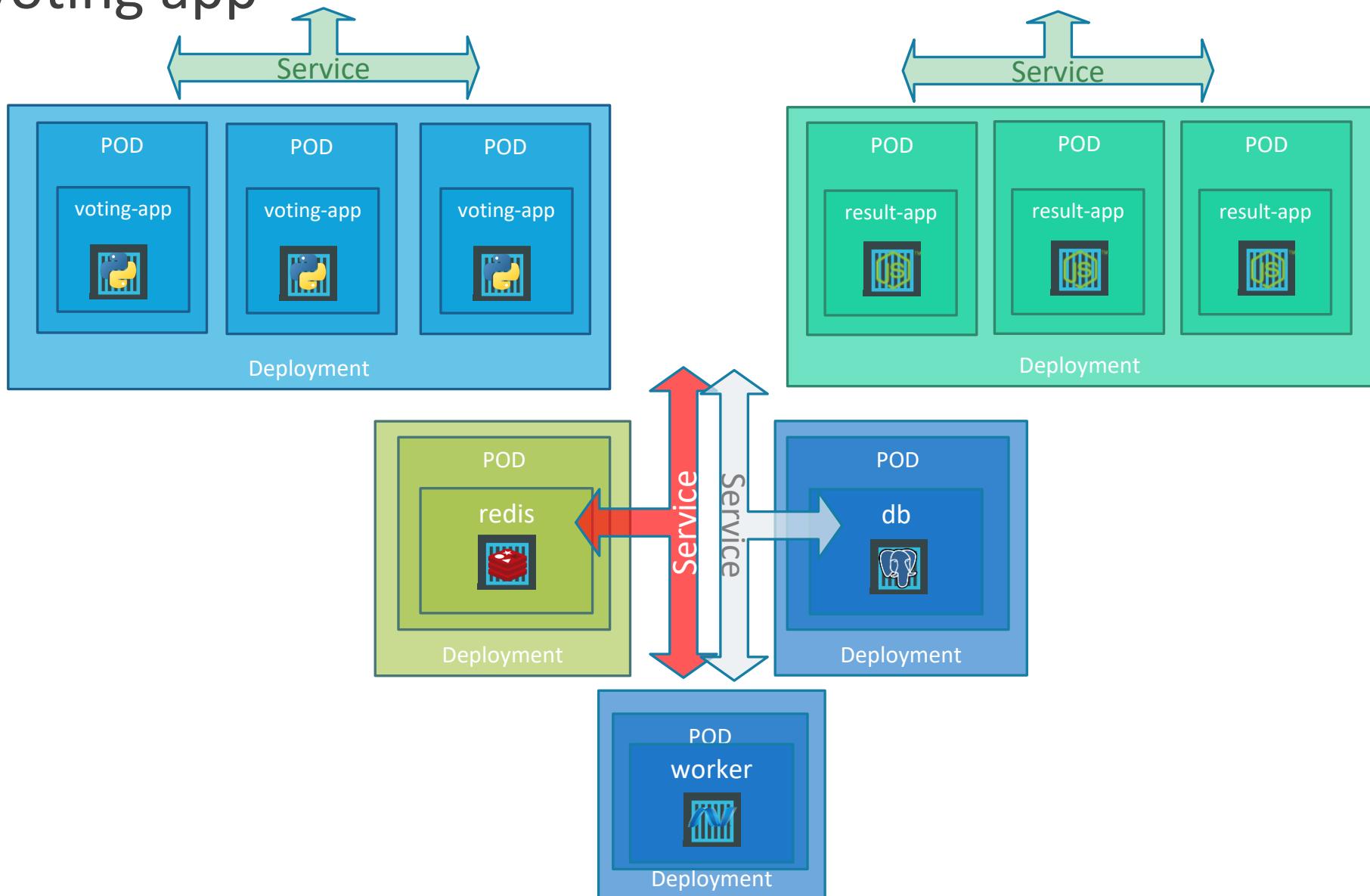
1. Deploy PODs
2. Create Services (ClusterIP)
 1. redis
 2. db
3. Create Services (NodePort)
 1. voting-app
 2. result-app



Example voting app



Example voting app





KodeKloud

Check out our full course on Kubernetes for the Absolute Beginners: <https://kode.wiki/43Sv88X>

Ansible For the Absolute Beginner



The RedHat Ansible for Beginners **Curriculum**

- Introduction to Ansible
- Setting up Ansible on VirtualBox
- Inventory Files
- Playbooks
- Variables
- Conditionals
- Loops
- Roles

Hands-On Exercises

Quiz Portal +

```
[ *****
ering Facts] *****

*****  
b1]: FAILED! => {"changed": false, "msg": "Unsupported  
parameters include: force, pesize, pv_options, pvs, state,  
retry, use: --limit @/home/thor/playbooks/create_vg.re  
*****  
: ok=1    changed=0    unreachable=0  
  
le-controller ~/playbooks$ ^C  
le-controller ~/playbooks$ vi create_vg.yml  
le-controller ~/playbooks$ ansible-playbook -i inventor  
[ *****  
ering Facts] *****
```

01 02 03 04 05

00:00

Under `~/playbooks/` directory create a playbook **create_vg.yml**. The playbook should create a new VG called **vg_data**. The playbook should run on node **web1**. Use the PV **/dev/vdb1** for the VG

Check

Hint

✗ Tasks not completed!

✓ Syntax Check



KodeKloud

Check out our full course on Ansible for the Absolute Beginners: <https://kode.wiki/3No1wef>

Ansible

Introduction

Why Ansible?



Provisioning



Configuration
Management



Continuous
Delivery



Application
Deployment



Security
Compliance



Scripts

- Time
- Coding Skills
- Maintenance



- Simple
- Powerful
- Agentless

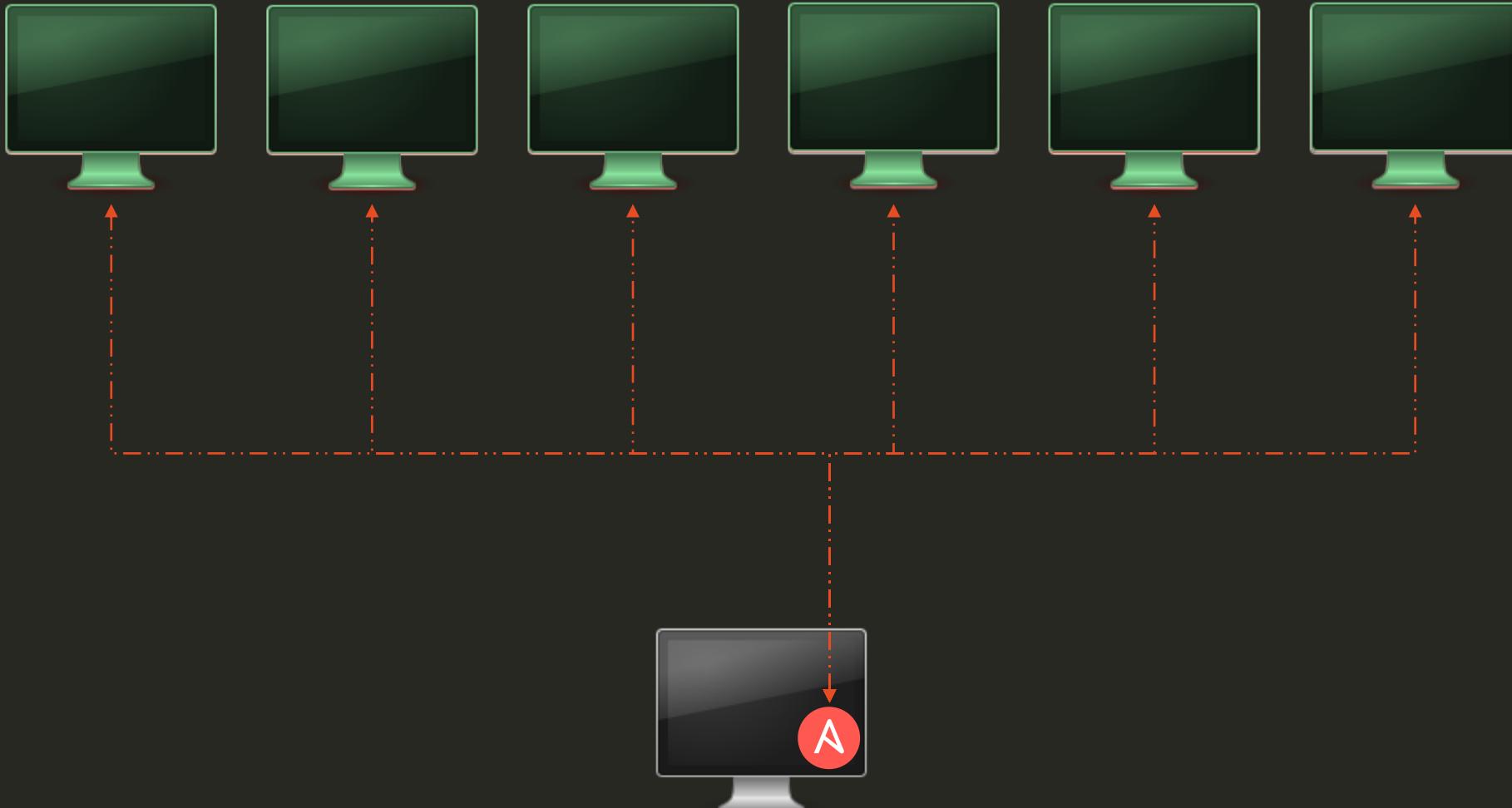
Scripts

vs Ansible Playbook

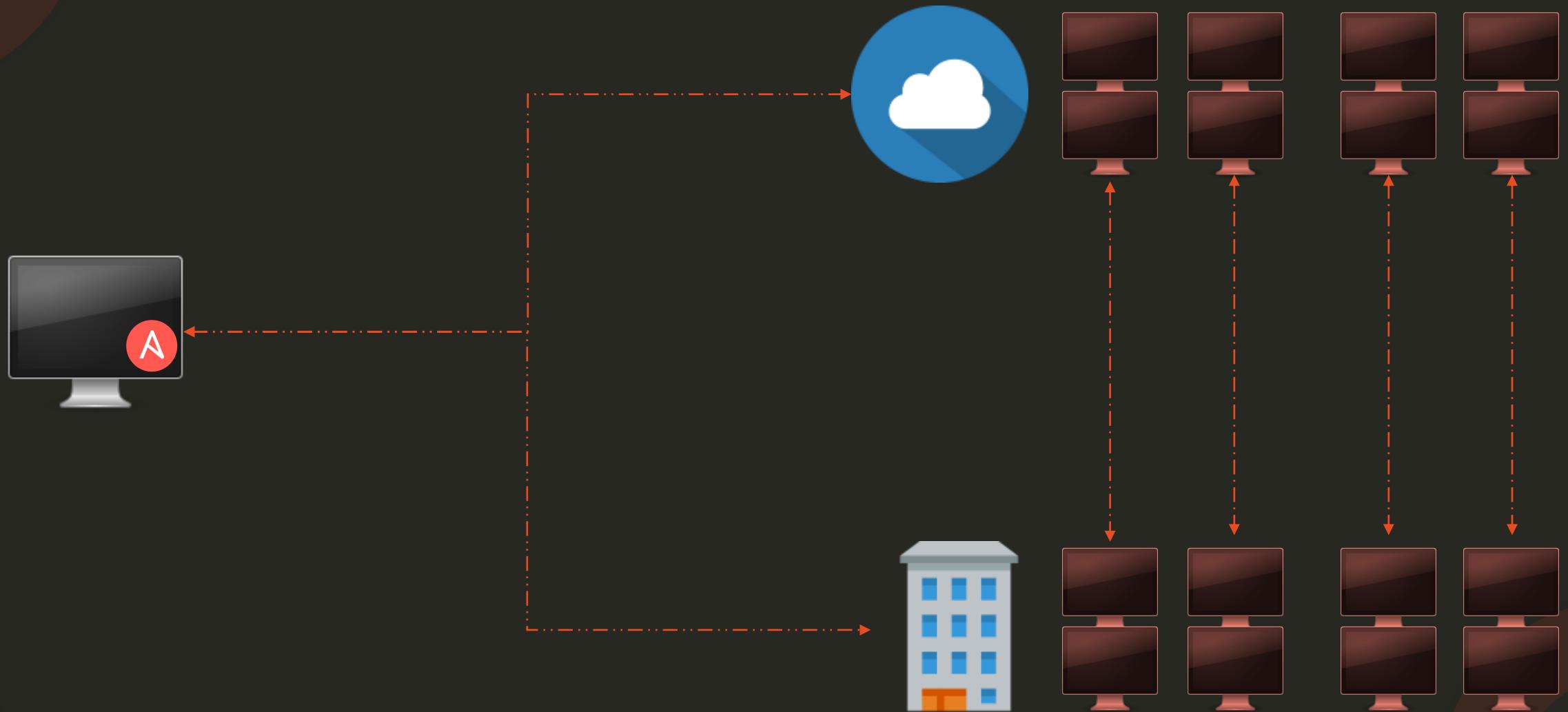
```
#!/bin/bash
# Script to add a user to Linux system
if [ $(id -u) -eq 0 ]; then
    $username=johndoe
    read -s -p "Enter password : " password
    egrep "^\$username" /etc/passwd >/dev/null
    if [ $? -eq 0 ]; then
        echo "$username exists!"
        exit 1
    else
        useradd -m -p $password $username
        [ $? -eq 0 ] && echo "User has been added
to system!" || echo "Failed to add a user!"
    fi
fi
```

```
- hosts: all_my_web_servers_in_DR
  tasks:
    - user:
        name: johndoe
```

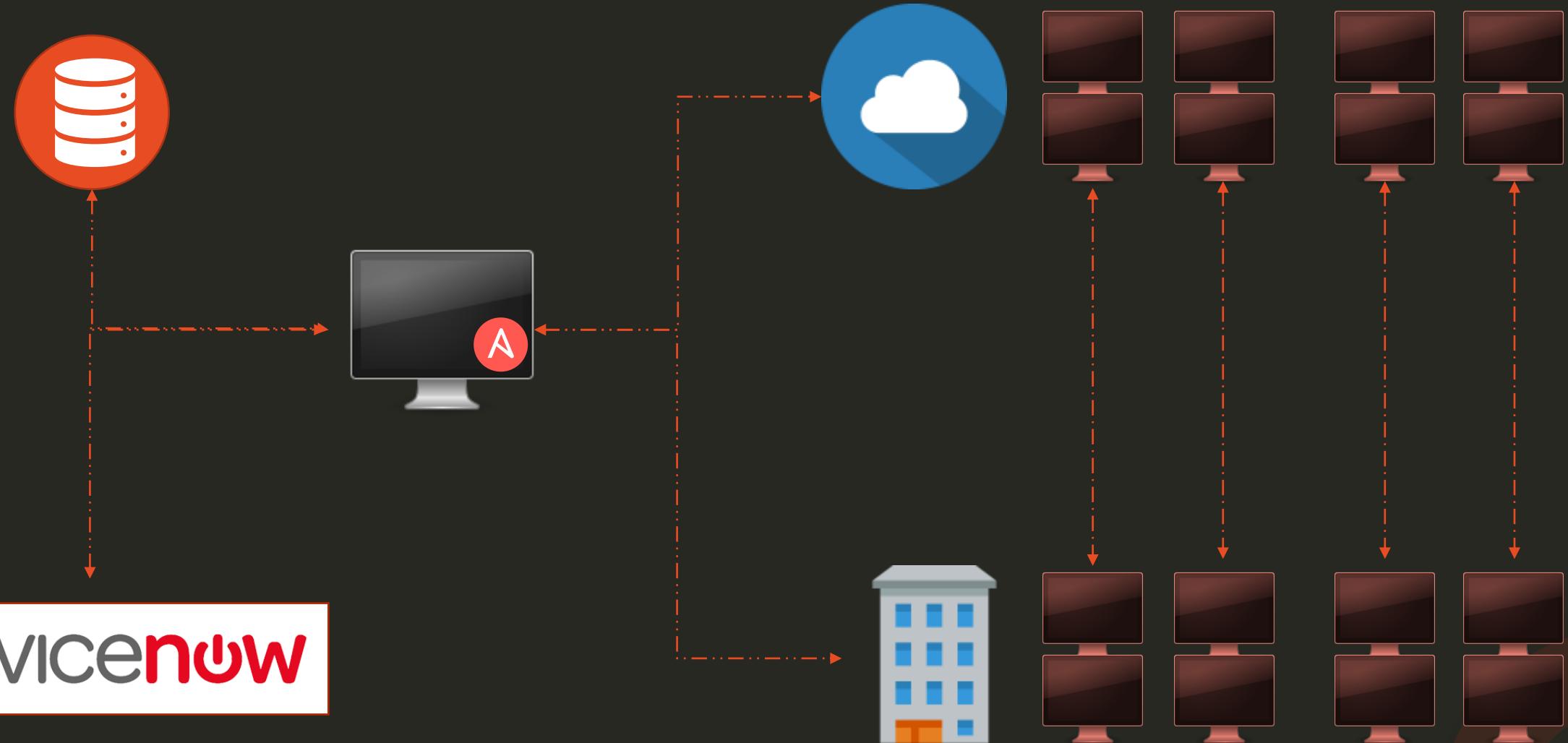
Use case example - Simple



Use case example - complex



Use case example - complex





KodeKloud

Check out our full course on Ansible for the Absolute Beginners: <https://kode.wiki/3No1wef>

Ansible

Install

Control Node



Redhat or CentOS – `$ sudo yum install ansible`



Fedora – `$ sudo dnf install ansible`



Ubuntu – `$ sudo apt-get install ansible`



PIP – `$ sudo pip install ansible`



Ansible Control
Machine

- Playbooks
- Inventory
- Modules

Additional Options:

- Install from source on GIT
- Build RPM yourself



Control Machine - Linux Only

Install Control Node on Redhat or CentOS



Redhat or CentOS – `$ sudo yum install ansible`

Install via PIP

Install pip if not present

```
$ sudo yum install epel-release
```

```
$ sudo yum install python-pip
```

Install Ansible using pip

```
$ sudo pip install ansible
```

Upgrade Ansible using pip

```
$ sudo pip install --upgrade ansible
```

Install Specific Version of Ansible using pip

```
$ sudo pip install ansible==2.4
```

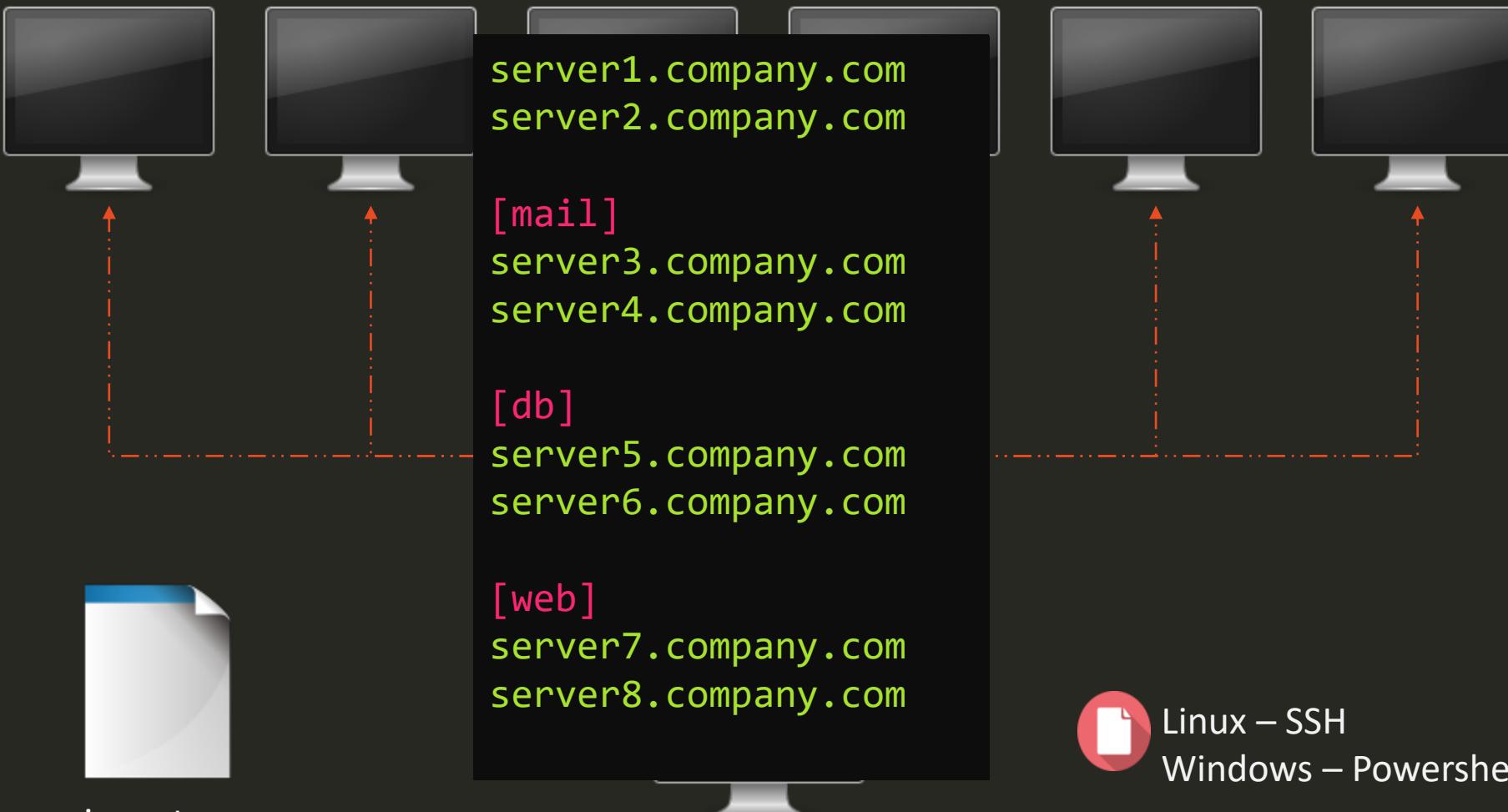


KodeKloud

Check out our full course on Ansible for the Absolute Beginners: <https://kode.wiki/3No1wef>

Ansible Inventory

inventory



Linux – SSH

Windows – Powershell Remoting



Agentless

Check out our full course on Ansible for the Absolute Beginners: <https://kode.wiki/3No1wef>

More on inventory files

```
#Sample Inventory File

server1 ansible_host=192.168.1.100
server2 ansible_host=192.168.1.101
server3 ansible_host=192.168.1.102
server4 ansible_host=192.168.1.103

ansible_connection=ssh      ansible_user=root
ansible_connection=winrm   ansible_user=admin
ansible_connection=ssh      ansible_ssh_pass=P@#%
ansible_connection=winrm

localhost ansible_connection=localhost
```



Inventory Parameters:

- ansible_connection – ssh/winrm/localhost
- ansible_port – 22/5986
- ansible_user – root/administrator
- ansible_ssh_pass - Password



Security: Ansible Vault

Check out our full course on Ansible for the Absolute Beginners: <https://kode.wiki/3No1wef>

Ansible Playbooks

Ansible playbooks

```
# Simple Ansible Playbook
```

- Run command1 on server1
- Run command2 on server2
- Run command3 on server3
- Run command4 on server4
- Run command5 on server5
- Run command6 on server6
- Run command7 on server7
- Run command8 on server8
- Run command9 on server9
- Restarting Server1
- Restarting Server2
- Restarting Server3
- Restarting Server4
- Restarting Server5
- Restarting Server6
- Restarting Server7

```
# Complex Ansible Playbook
```

- Deploy 50 VMs on Public Cloud
- Deploy 50 VMs on Private Cloud
- Provision Storage to all VMs
- Setup Network Configuration on Private VMs
- Setup Cluster Configuration
- Configure Web server on 20 Public VMs
- Configure DB server on 20 Private VMs
- Setup Loadbalancing between web server VMs
- Setup Monitoring components
- Install and Configure backup clients on VMs
- Update CMDB database with new VM Information

Playbook

- Playbook – A single YAML file
 - Play – Defines a set of activities (tasks) to be run on hosts
 - Task – An action to be performed on the host
 - Execute a command
 - Run a script
 - Install a package
 - Shutdown/Restart



YAML format

playbook.yml

```
-  
  name: Play 1  
  hosts: localhost  
  tasks:  
    - name: Execute command 'date'  
      command: date  
  
    - name: Execute script on server  
      script: test_script.sh  
  
    - name: Install httpd service  
      yum:  
        name: httpd  
        state: present  
  
    - name: Start web server  
      service:  
        name: httpd  
        state: started
```

Playbook format

playbook.yml

```
name: Play 1
hosts: localhost
tasks:
  - name: Execute command 'date'
    command: date

  - name: Execute script on server
    script: test_script.sh

name: Play 2
hosts: localhost
tasks:
  - name: Install web service
    yum:
      name: httpd
      state: present

  - name: Start web server
    service:
      name: httpd
      state: started
```

Hosts

playbook.yml

```
-  
  name: Play 1  
  hosts: localhost  
  tasks:  
    - name: Execute command 'date'  
      command: date  
  
    - name: Execute script on server  
      script: test_script.sh  
  
    - name: Install httpd service  
      yum:  
        name: httpd  
        state: present  
  
    - name: Start web server  
      service:  
        name: httpd  
        state: started
```

inventory

localhost

server1.company.com

server2.company.com

[mail]

server3.company.com

server4.company.com

[db]

server5.company.com

server6.company.com

[web]

server7.company.com

server8.company.com

module

playbook.yml

```
-  
  name: Play 1  
  hosts: localhost  
  tasks:  
    - name: Execute command 'date'  
      command: date  
  
    - name: Execute script on server  
      script: test_script.sh  
  
    - name: Install httpd service  
      yum:  
        name: httpd  
        state: present  
  
    - name: Start web server  
      service:  
        name: httpd  
        state: started
```



ansible-doc -l

Run

- Execute Ansible Playbook
- Syntax: `ansible-playbook <playbook file name>`



`ansible-playbook playbook.yml`



`ansible-playbook --help`



KodeKloud

Check out our full course on Ansible for the Absolute Beginners: <https://kode.wiki/3No1wef>

Ansible Modules

modules

- System
- Commands
- Files
- Database
- Cloud
- Windows
- More..

- Win_copy
- Win_command
- Win_domain
- Win_file
- Win_iis_website
- Win_msg
- Win_msi
- Win_package
- Win_ping
- Win_path
- Win_robocopy
- Win_regedit
- Win_shell
- Win_service
- Win_user
- And more

command

Executes a command on a remote node

parameter	comments
chdir	cd into this directory before running the command
creates	a filename or (since 2.0) glob pattern, when it already exists, this step will not be run.
executable	change the shell used to execute the command. Should be an absolute path to the executable.
free_form	the command module takes a free form command to run. There is no parameter actually named 'free form'. See the examples!
removes	a filename or (since 2.0) glob pattern, when it does not exist, this step will not be run.
warn (added in 1.8)	if command warnings are on in ansible.cfg, do not warn about this particular line if set to no/false.

playbook.yml

```
- name: Play 1
hosts: localhost
tasks:
  - name: Execute command 'date'
    command: date

  - name: Display resolv.conf contents
    command: cat /etc/resolv.conf

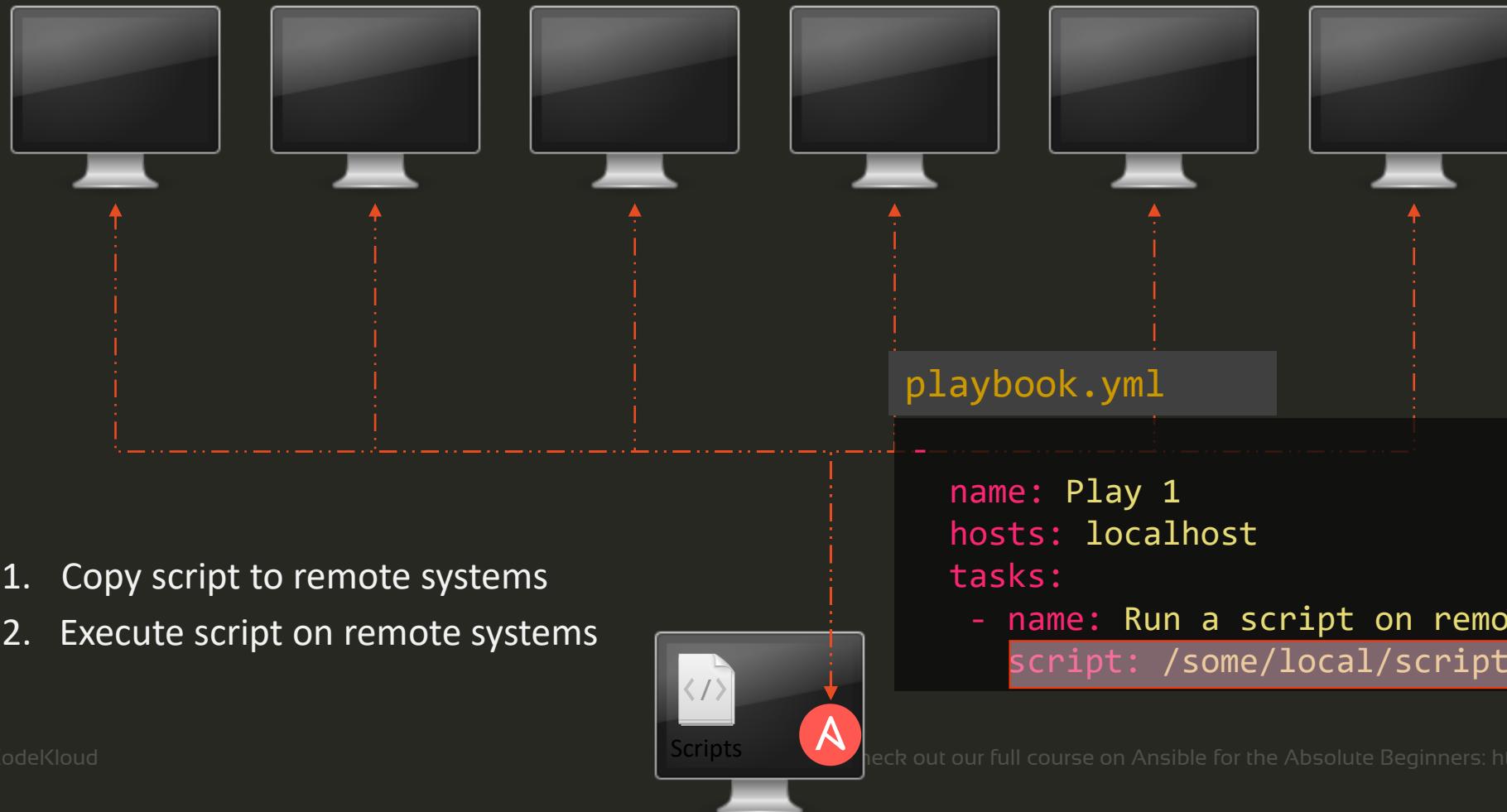
  - name: Display resolv.conf contents
    command: cat resolv.conf chdir=/etc

  - name: Display resolv.conf contents
    command: mkdir /folder creates=/folder
```

```
- name: Copy file from source to destination
copy: src=/source_file dest=/destination
```

script

- Runs a local script on a remote node after transferring it



Service

- Manage Services – Start, Stop, Restart

playbook.yml

```
-  
  name: Start Services in order  
  hosts: localhost  
  tasks:  
    - name: Start the database service  
      service: name=postgresql state=started  
  
    - name: Start the httpd service  
      service: name=httpd state=started  
  
    - name: Start the nginx service  
      service:  
        name: nginx  
        state: started
```

playbook.yml

```
-  
  name: Start Services in order  
  hosts: localhost  
  tasks:  
    - name: Start the database service  
      service:  
        name: postgresql  
        state: started
```

idempotency

Why “started” and not “start”?

“Start” the service httpd

“Started” the service httpd

Ensure service httpd is started

If httpd is not already started => start it

If httpd is already started, =>do nothing

Idempotency

An operation is idempotent if the result of performing it once is exactly the same as the result of performing it repeatedly without any intervening actions.

lineinfile

- Search for a line in a file and replace it or add it if it doesn't exist.

```
/etc/resolv.conf
```

```
nameserver 10.1.250.1
nameserver 10.1.250.2
```

```
nameserver 10.1.250.10
```

```
playbook.yml
```

```
- name: Add DNS server to resolv.conf
  hosts: localhost
  tasks:
    - lineinfile:
        path: /etc/resolv.conf
        line: 'nameserver 10.1.250.10'
```

```
script.sh
```

```
#Sample script
```

```
echo "nameserver 10.1.250.10" >> /etc/resolv.conf
```

```
/etc/resolv.conf
```

```
nameserver 10.1.250.1
nameserver 10.1.250.2
nameserver 10.1.250.10
```

```
/etc/resolv.conf
```

```
nameserver 10.1.250.1
nameserver 10.1.250.2
nameserver 10.1.250.10
nameserver 10.1.250.10
nameserver 10.1.250.10
```



KodeKloud

Check out our full course on Ansible for the Absolute Beginners: <https://kode.wiki/3No1wef>

Ansible Variables

Variable

- Stores information that varies with each host

inventory

```
Web1 ansible_host=server1.company.com ansible_connection=ssh    ansible_ssh_pass=P@ssW
db    ansible_host=server2.company.com ansible_connection=winrm ansible_ssh_pass=P@s
Web2 ansible_host=server3.company.com ansible_connection=ssh    ansible_ssh_pass=P@ssW
```

Playbook.yml

```
- 
  name: Add DNS server to resolv.conf
  hosts: localhost
  tasks:
    dns_lineinfile10.1.250.10
      path: /etc/resolv.conf
      line: 'nameserver 10.1.250.10'
```

variables

```
variable1: value1
variable2: value2
```

Using variables

Playbook.yml

```
-  
  name: Add DNS server to resolv.conf  
  hosts: localhost  
  vars:  
    dns_server: 10.1.250.10  
  tasks:  
    - lineinfile:  
        path: /etc/resolv.conf  
        line: 'nameserver {{ dns_server }}'
```

```
- name: Set Firewall Configurations
hosts: web
tasks:
- firewalld:
  service: https
  permanent: true
  state: enabled

- firewalld:
  port: {{ http_port }}/tcp
  permanent: true
  state: disabled

- firewalld:
  port: {{ snmp_port }}/udp
  permanent: true
  state: disabled

- firewalld:
  source: {{ inter_ip_range }}/24
  Zone: internal
  state: enabled
```

```
#Sample Inventory File
Web http_port=      snmp_port=      inter_ip_range=
```

```
#Sample variable File - web.yml
http_port: 8081
snmp_port: 161-162
inter_ip_range: 192.0.2.0
```

{}
}

Jinja2 Templating



source: {{ inter_ip_range }}



source: '{{ inter_ip_range }}'

source: SomeThing{{ inter_ip_range }}SomeThing

Check out our full course on Ansible for the Absolute Beginners: <https://kode.wiki/3Nolwef>

Ansible Loops

LOOPS

```
- name: Create users
hosts: localhost
tasks:
- user: name=joe item }' state=present
- user: name=george state=present
- user: name=ravi state=present
- user: name=mani state=present
- user: name=kiran state=present
- user: name=jazlan state=present
- user: name=emaan state=present
- user: name=mazin state=present
- user: name=izaan state=present
- user: name=mike state=present
- user: name=menaal state=present
- user: name=shoeb state=present
- user: name=rani state=present
    - shoeb
    - rani
```

LOOPS - Visualize

```
- name: Create users
  hosts: localhost
  tasks:
    - user: name='{{ item }}'      state=present
      loop:
        - joe
        - george
        - ravi
        - mani
        - kiran
        - jazlan
        - emaan
        - mazin
        - izaan
        - mike
        - menaal
        - shoeb
        - rani
```

```
- name: Create users
  hosts: localhost
  tasks:
    - var: item=
      user: name="{{ item }}"      state=present
    - var: item=
      user: name="{{ item }}"      state=present
```

Checkout our full course on Ansible for Absolute Beginners: <https://rootsh.wiki/known>

LOOPS - Visualize

```
- name: Create users
  hosts: localhost
  tasks:
    - user: name='{{ item }}'      state=present
      loop:
        - joe
        - george
        - ravi
        - mani
        - kiran
        - jazlan
        - emaan
        - mazin
        - izaan
        - mike
        - menaal
        - shoeb
        - rani
```

```
- name: Create users
  hosts: localhost
  tasks:
    - var: item=joe
      user: name="{{ item }}"      state=present
    - var: item=george
      user: name="{{ item }}"      state=present
    - var: item=ravi
      user: name="{{ item }}"      state=present
    - var: item=mani
      user: name="{{ item }}"      state=present
    - var: item=kiran
      user: name="{{ item }}"      state=present
    - var: item=jazlan
      user: name="{{ item }}"      state=present
    - var: item=emaan
      user: name="{{ item }}"      state=present
    - var: item=mazin
      user: name="{{ item }}"      state=present
    - var: item=izaan
      user: name="{{ item }}"      state=present
```

Checkout our full course on Ansible for Absolute Beginners: <https://rootshub.io/course/ansible-for-beginners>

LOOPS - Visualize

```
- name: Create users
  hosts: localhost
  tasks:
    - user: name '{{ ??? }}' state=present uid='{{ ? }}'
      loop:
        - name: joe
          uid: 1010
        - name: george
          uid: 1011
        - name: ravi
          uid: 1012
        - name: mani
          uid: 1013
        - name: kiran
          uid: 1014
        - name: jazlan
          uid: 1015
        - name: emaan
          uid: 1016
        - name: mazin
          uid: 1017
        - name: izaan
          uid: 1018
        - name: mike
```

```
- name: Create users
  hosts: localhost
  tasks:
    - var: item=joe
      user: name="{{ item }}" state=present
    - var: item=george
      user: name="{{ item }}" state=present
    - var: item=ravi
      user: name="{{ item }}" state=present
    - var: item=mani
      user: name="{{ item }}" state=present
    - var: item=kiran
      user: name="{{ item }}" state=present
    - var: item=jazlan
      user: name="{{ item }}" state=present
    - var: item=emaan
      user: name="{{ item }}" state=present
    - var: item=mazin
      user: name="{{ item }}" state=present
    - var: item=izaan
      user: name="{{ item }}" state=present
```

Checkout our full course on Ansible for Absolute Beginners: <https://realpython.com/learn/>

LOOPS - Visualize

```
- name: Create users
hosts: localhost
tasks:
- user: name '{{ ??? }}' state=present uid='{{ ? }}'
loop:
```

- name: joe
uid: 1010
- name: george
uid: 1011
- name: ravi
uid: 1012
- name: mani
uid: 1013
- name: kiran
uid: 1014
- name: jazlan
uid: 1015
- name: emaan
uid: 1016
- name: mazin
uid: 1017
- name: izaan
uid: 1018
- name: mike

```
- name: Create users
hosts: localhost
tasks:
- var:
  item:  
  
    user: name="{{ ??? }}" state=present uid="{{?}}"  
  
- var:
  item:  
  
    user: name="{{ ??? }}" state=present uid="{{?}}"  
  
- var:
  item:  
  
    user: name="{{ ??? }}" state=present uid="{{?}}"  
  
- var:
  item:  
  
    user: name="{{ ??? }}" state=present uid="{{?}}"  
  
- var:
  item:  
  
    user: name="{{ ??? }}" state=present uid="{{?}}"  
  
- var:
  item:  
  
    user: name="{{ ??? }}" state=present uid="{{?}}"
```

Checkout our full course on Ansible for the Absolute Beginners: <https://kodewiki/3Nolwef>

```
user: name="{{ ??? }}" state=present uid="{{?}}"
```

LOOPS - Visualize

```
-  
  name: Create users  
  hosts: localhost  
  tasks:  
    - user: name '{{ ??? }}' state=present uid='{{ ? }}'  
      loop:  
        - name: joe  
          uid: 1010  
        - name: george  
          uid: 1011  
        - name: ravi  
          uid: 1012  
        - name: mani  
          uid: 1013  
        - name: kiran  
          uid: 1014  
        - name: jazlan  
          uid: 1015  
        - name: emaan  
          uid: 1016  
        - name: mazin  
          uid: 1017  
        - name: izaan  
          uid: 1018  
        - name: mike
```

```
-  
  name: Create users  
  hosts: localhost  
  tasks:  
    - var:  
        item:  
          name: joe  
          uid: 1010  
          user: name='{{ item.name }}' state=present uid='{{ item.uid }}'  
    - var:  
        item:  
          name: george  
          uid: 1011  
          user: name='{{ item.name }}' state=present uid='{{ item.uid }}'  
    - var:  
        item:  
          name: ravi  
          uid: 1012  
          user: name='{{ item.name }}' state=present uid='{{ item.uid }}'  
    - var:  
        item:  
          name: mani  
          uid: 1013  
          user: name='{{ item.name }}' state=present uid='{{ item.uid }}'
```

Check out our full course on Ansible for the Absolute Beginners: <https://kodewiki/3Nolwef>

LOOPS - Visualize

```
- name: Create users
  hosts: localhost
  tasks:
    - user: name='{{ item.name }}' state=present uid='{{ item.uid }}'
      loop:
        - name: joe      - { name: joe, uid: 1010 }
          uid: 1010
        - name: george   - { name: george, uid: 1011 }
          uid: 1011
        - name: ravi     - { name: ravi, uid: 1012 }
          uid: 1012
        - name: mani     - { name: mani, uid: 1013 }
          uid: 1013
        - name: kiran    - { name: kiran, uid: 1014 }
          uid: 1014
        - name: jazlan   - { name: jazlan, uid: 1015 }
          uid: 1015
        - name: emaan    - { name: emaan, uid: 1016 }
          uid: 1016
        - name: mazin    - { name: mazin, uid: 1017 }
          uid: 1017
        - name: izaan    - { name: izaan, uid: 1018 }
          uid: 1018
        - name: mike     - { name: mike, uid: 1019 }
```

```
- name: Create users
  hosts: localhost
  tasks:
    - var:
        item:
          name: joe
          uid: 1010
          user: name='{{ item.name }}' state=present uid='{{ item.uid }}'
    - var:
        item:
          name: george
          uid: 1011
          user: name='{{ item.name }}' state=present uid='{{ item.uid }}'
    - var:
        item:
          name: ravi
          uid: 1012
          user: name='{{ item.name }}' state=present uid='{{ item.uid }}'
    - var:
        item:
          name: mani
          uid: 1013
          user: name='{{ item.name }}' state=present uid='{{ item.uid }}'
```

With_*

```
- name: Create users
hosts: localhost
tasks:
  - user: name='{{ item }}'    state=present
loop:
  - joe
  - george
  - ravi
  - mani
```

```
- name: Create users
hosts: localhost
tasks:
  - user: name='{{ item }}'    state=present
    with_items:
      - joe
      - george
      - ravi
      - mani
```

With_*

```
-  
  name: Create users  
  hosts: localhost  
  tasks:  
    - user: name='{{ item }}' state=present  
      with_items:  
        - joe  
        - george  
        - ravi  
        - mani
```

```
-  
  name: Get from multiple URLs  
  hosts: localhost  
  tasks:  
    - debug: var=item  
      with_url:  
        - "https://site1.com/get-servers"  
        - "https://site2.com/get-servers"  
        - "https://site3.com/get-servers"
```

```
-  
  name: View Config Files  
  hosts: localhost  
  tasks:  
    - debug: var=item  
      with_file:  
        - "/etc/hosts"  
        - "/etc/resolv.conf"  
        - "/etc/ntp.conf"
```

```
-  
  name: Check multiple mongodbs  
  hosts: localhost  
  tasks:  
    - debug: msg="DB={{ item.database }} PID={{ item.pid }}"  
      with_mongodb:  
        - database: dev  
          connection_string: "mongodb://dev.mongo/"  
        - database: prod  
          connection_string: "mongodb://prod.mongo/"
```

With_*

with_items
with_file
with_url
with_mongodb

with_dict
with_etcd
with_env
with_filetree
With_ini
With_inventory_hostnames
With_k8s
With_manifold
With_nested
With_nios
With_openshift
With_password
With_pipe
With_rabbitmq

With_redis
With_sequence
With_skydive
With_subelements
With_template
With_together
With_varnames



KodeKloud

Check out our full course on Ansible for the Absolute Beginners: <https://kode.wiki/3No1wef>

Ansible Conditionals

```
---  
- name: Install NGINX  
hosts: debian_hosts  
tasks:  
- name: Install NGINX on Debian  
  apt:  
    name: nginx  
    state: present
```

```
---  
- name: Install NGINX  
hosts: redhat_hosts  
tasks:  
- name: Install NGINX on Redhat  
  yum:  
    name: nginx  
    state: present
```

Conditional - when

```
---
- name: Install NGINX
  hosts: all
  tasks:
    - name: Install NGINX on Debian
      apt:
        name: nginx
        state: present
      when: ansible_os_family == "Debian"

    - name: Install NGINX on Redhat
      yum:
        name: nginx
        state: present
      when: ansible_os_family == "RedHat"
```

Operator - or

```
---
- name: Install NGINX
  hosts: all
  tasks:
    - name: Install NGINX on Debian
      apt:
        name: nginx
        state: present
        when: ansible_os_family == "Debian"

    - name: Install NGINX on Redhat
      yum:
        name: nginx
        state: present
        when: ansible_os_family == "RedHat" or
              ansible_os_family == "SUSE"
```

Operator - and

```
---
```

- name: Install NGINX
 - hosts: all
 - tasks:
 - name: Install NGINX on Debian
 - apt:
 - name: nginx
 - state: present
 - when: ansible_os_family == "Debian" and ansible_distribution_version == "16.04"
 - name: Install NGINX on Redhat
 - yum:
 - name: nginx
 - state: present
 - when: ansible_os_family == "RedHat" or ansible_os_family == "SUSE"

Conditionals in Loops

```
---
- name: Install NGINX
  hosts: all
  tasks:
    - name: Install NGINX on Debian
      apt:
        name: nginx
        state: present
```

Conditionals in Loops

```
---
- name: Install Softwares
  hosts: all
  vars:
    packages:
      - name: nginx
        required: True
      - name: mysql
        required : True
      - name: apache
        required : False
  tasks:
    - name: Install "{{ item.name }}" on Debian
      apt:
        name: "{{ item.name }}"
        state: present
      loop: "{{ packages }}
```

Conditionals in Loops

```
---
```

```
- name: Install Softwares
hosts: all
vars:
  packages:
    - name: nginx
      required: True
    - name: mysql
      required : True
    - name: apache
      required : False
tasks:
- name: Install "{{ item.name }}" on Debian
  apt:
    name: "{{ item.name }}"
    state: present
  loop: "{{ packages }}
```

```
- name: Install "{{ item.name }}" on Debian
  vars:
    item:
      name: nginx
      required: True
  apt:
    name: "{{ item.name }}"
    state: present
    when: item.required == True
- name: Install "{{ item.name }}" on Debian
  vars:
    item:
      name: mysql
      required: True
  apt:
    name: "{{ item.name }}"
    state: present
    when: item.required == True
- name: Install "{{ item.name }}" on Debian
  vars:
    item:
      name: apache
      required: False
  apt:
    name: "{{ item.name }}"
    state: present
    when: item.required == True
```

Conditionals in Loops

```
---
- name: Install Softwares
  hosts: all
  vars:
    packages:
      - name: nginx
        required: True
      - name: mysql
        required : True
      - name: apache
        required : False
  tasks:
    - name: Install "{{ item.name }}" on Debian
      apt:
        name: "{{ item.name }}"
        state: present
      when: item.required == True
      loop: "{{ packages }}
```

Conditionals & Register

```
- name: Check status of a service and email if its down
hosts: localhost
tasks:
  - command: service httpd status
    register: result

  - mail:
      to: admin@company.com
      subject: Service Alert
      body: Httpd Service is down
      when: result.stdout.find('down') != -1
```

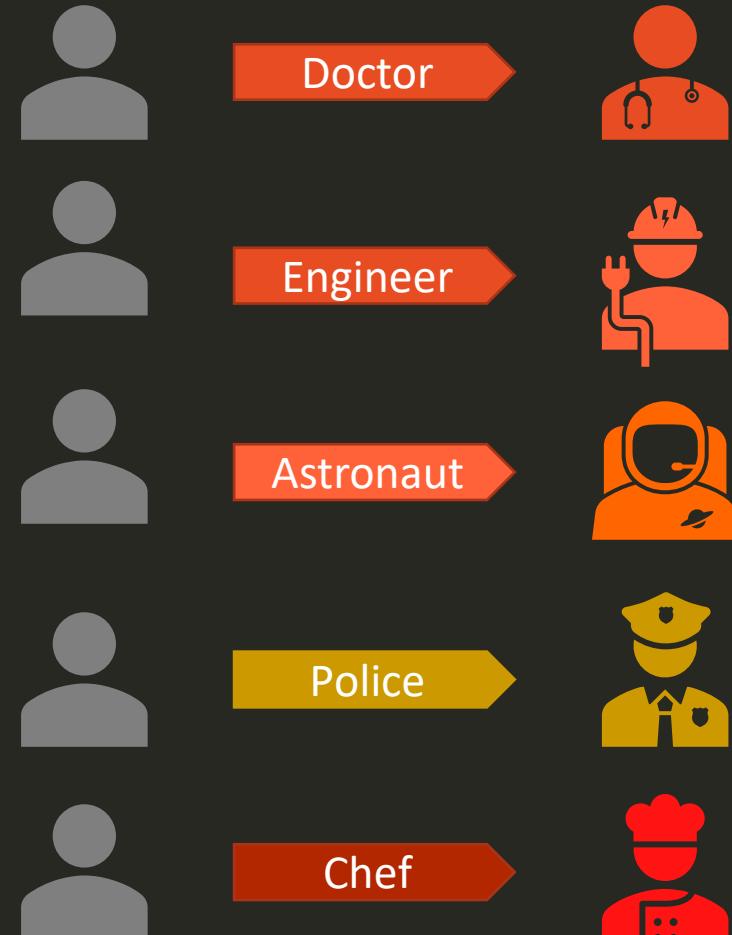


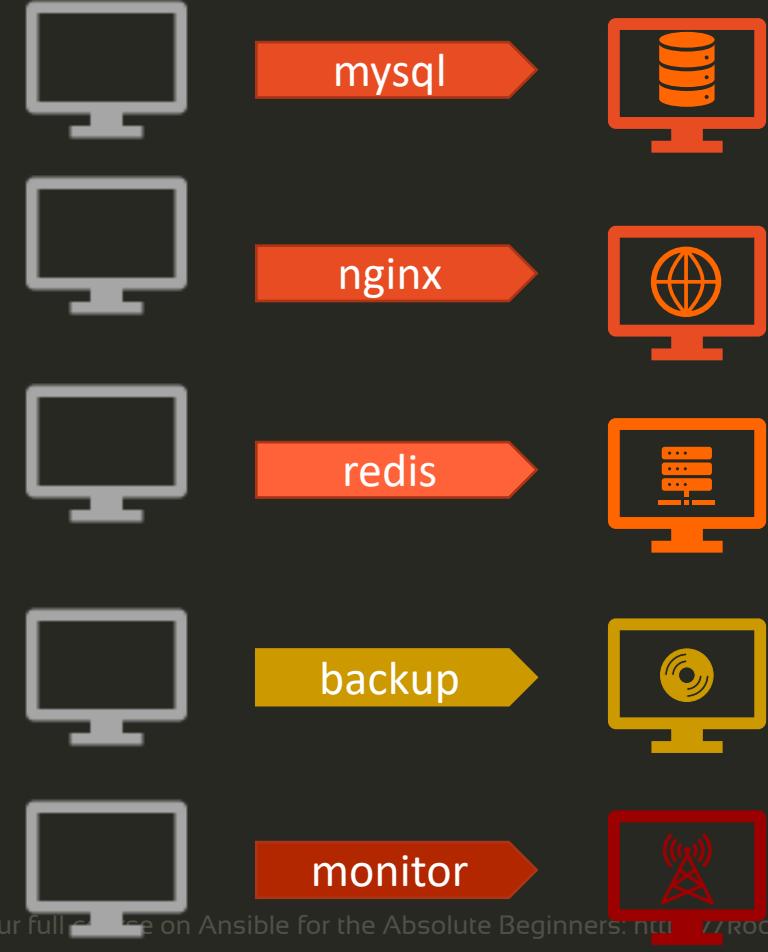
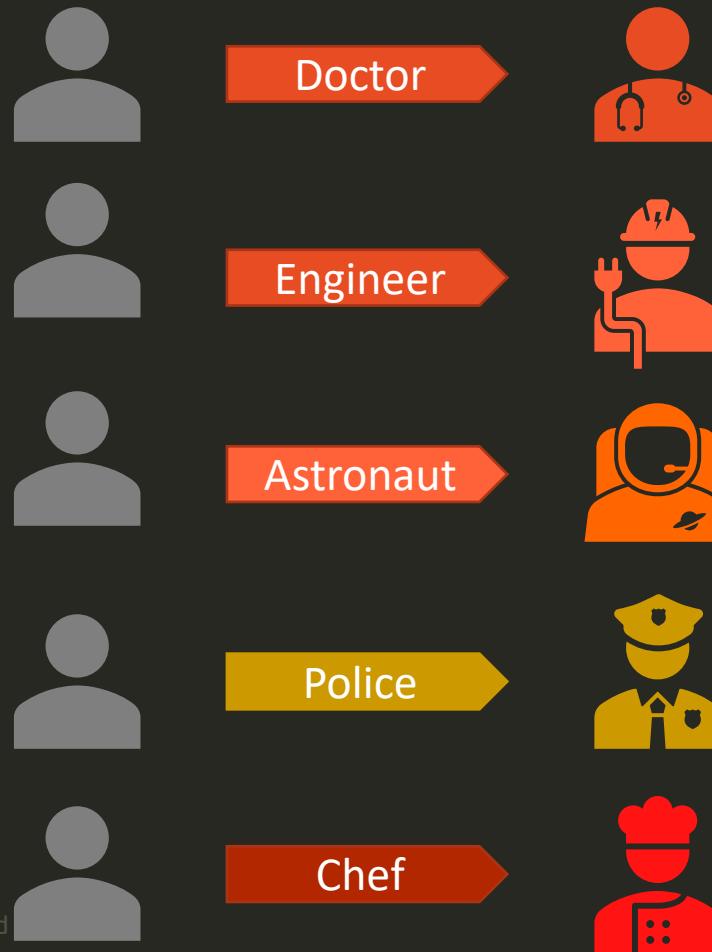
KodeKloud

Check out our full course on Ansible for the Absolute Beginners: <https://kode.wiki/3No1wef>

Ansible

Roles







Doctor

- Go to medical school
- Earn medical degree
- Complete Residency Program
- Obtain License



Engineer

- Go to engineering school
- Earn bachelor's degree
- Gain field experience
- Gain postgraduate degree



mysql

- Installing Pre-requisites
- Installing mysql packages
- Configuring mysql service
- Configuring database and users



nginx

- Installing Pre-requisites
- Installing nginx packages
- Configuring nginx service
- Configuring custom web pages



```
- name: Install and Configure MySQL
hosts: db-server
tasks:
  - name: Install Pre-Requisites
    yum: name=pre-req-packages state=present
  - name: Install MySQL Packages
    yum: name=mysql state=present
  - name: Start MySQL Service
    service: name=mysql state=started
  - name: Configure Database
    mysql_db: name=db1 state=present
```



mysql



- Installing Pre-requisites
- Installing mysql packages
- Configuring mysql service
- Configuring database and users



nginx



- Installing Pre-requisites
- Installing nginx packages
- Configuring nginx service
- Configuring custom web pages



Re-Use



mysql



- Installing Pre-requisites
- Installing mysql packages
- Configuring mysql service
- Configuring database and users

```
- name: Install and Configure MySQL
hosts: db-server1.....db-server100
roles:
  - mysql
```

MySQL-Role

tasks:

- name: Install Pre-Requisites
yum: name=pre-req-packages state=present
- name: Install MySQL Packages
yum: name=mysql state=present
- name: Start MySQL Service
service: name=mysql state=started
- name: Configure Database
mysql_db: name=db1 state=present



MySQL-Role

tasks

```
tasks:
  - name: Install Pre-Requisites
    yum: name=pre-req-packages state=present

  - name: Install MySQL Packages
    yum: name=mysql state=present

  - name: Start MySQL Service
    service: name=mysql state=started

  - name: Configure Database
    mysql_db: name=db1 state=present
```

vars

```
mysql_packages:
  - mysql
  - mysql-server
db_config:
  db_name: db1
```

handlers

defaults

```
mysql_user_name: root
mysql_user_password: root
```

templates



ansistrano

rollback

Ansible role to rollback scripting applications like PHP, Python, Ruby, etc. in a Capistrano style



cloud web

build passing

! 2.3 / 5 Score 61691 Downloads

Last Imported: 12 days ago



andrewrothst...

terraform

terraform role



cloud infrastructure terraform

✓ 4.2 / 5 Score 59591 Downloads

Last Imported: 8 days ago



sbaerlocher

do-agent

Cross-distro installation of the DigitalOcean monitoring agent



cloud monitoring

build passing

42166 Downloads

Last Imported: a year ago



CyVerse-Ansible

ez

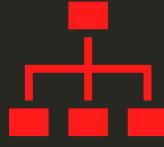
This role sets up the ez cli and other convenience functions commands by placing bash scripts into the /etc/profile.d of a system.



ansible bash cloud cyverse shell

35349 Downloads

Last Imported: 2 years ago



Organize



Re-Use



Share

```
$ ansible-galaxy init mysql
```



mysql



README.md



templates



tasks



handlers



vars



defaults



meta



my-playbook



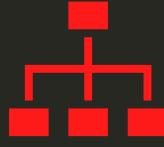
playbook.yml



roles

playbook.yml

- name: Install and Configure MySQL
- hosts: db-server
- roles:
 - mysql



Organize



Re-Use



Share

```
$ ansible-galaxy init mysql
```



mysql



README.md



templates



tasks



handlers



vars



defaults



meta



my-playbook



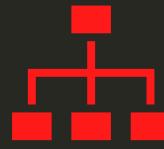
playbook.yml



roles

playbook.yml

```
- name: Install and Configure MySQL
  hosts: db-server
  roles:
    - mysql
```



Organize

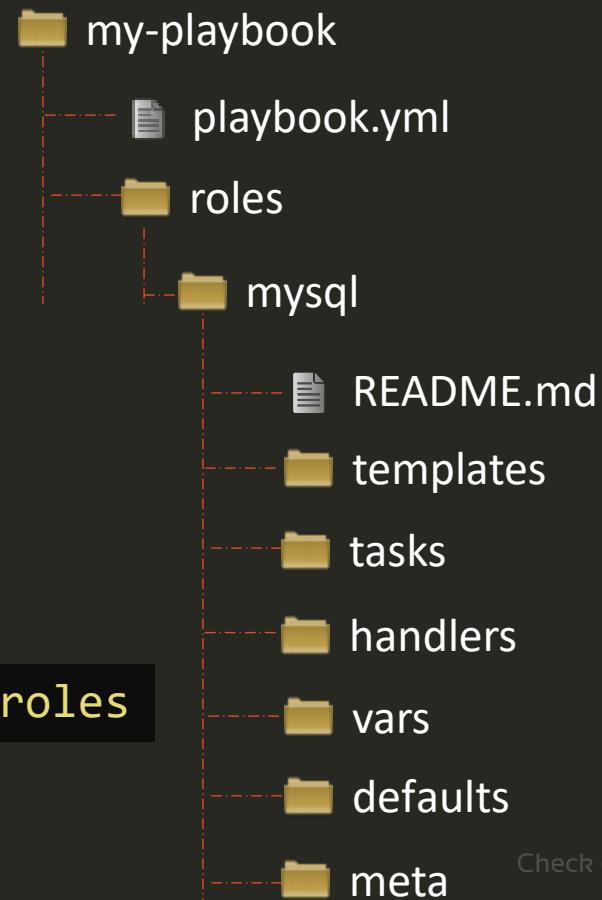


Re-Use



Share

```
$ ansible-galaxy init mysql
```

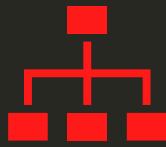


playbook.yml

```
- name: Install and Configure MySQL
  hosts: db-server
  roles:
    - mysql
```

```
/etc/ansible/ansible.cfg
```

```
roles_path = /etc/ansible/roles
```



Organize

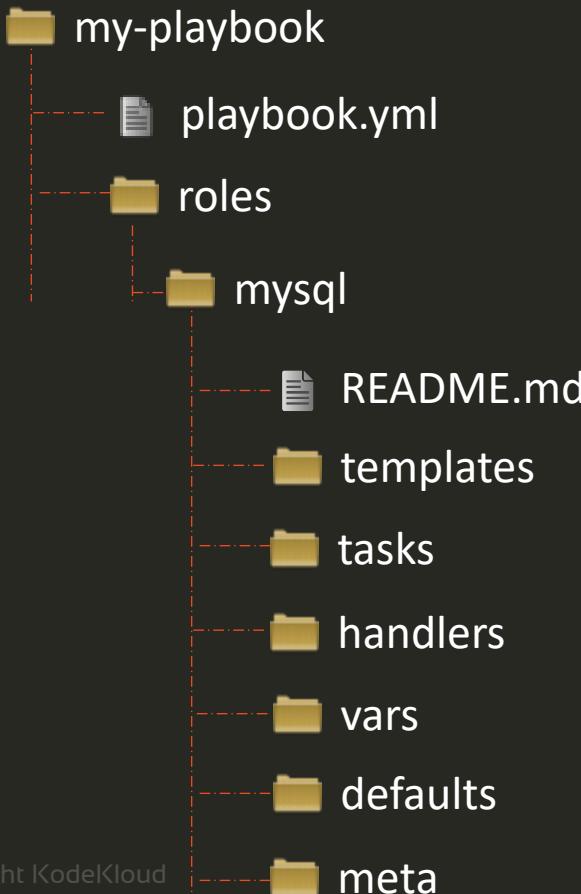


Re-Use



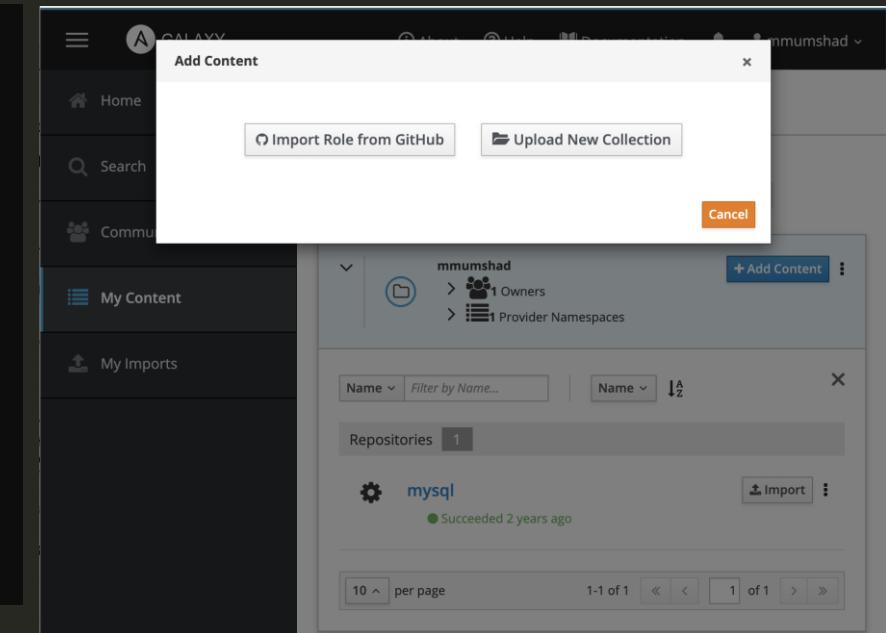
Share

```
$ ansible-galaxy init mysql
```



playbook.yml

```
- name: Install and Configure MySQL
  hosts: db-server
  roles:
    - mysql
```



Find Roles

Search mysql

Type Filter by Collection or Role... Best Match ↓ 288 Results Active filters: Tag: database × Clear All Filters

Roles 288

mysql
MySQL server for RHEL/CentOS and Debian/Ubuntu.
geerlingguy build passing 3.2 / 5 Score 512737 Downloads Last Imported: 5 days ago
Tags: database db mariadb mysql sql

php-mysql
PHP MySQL support for Linux.
geerlingguy build passing 5 / 5 Score 133181 Downloads Last Imported: 3 days ago
Tags: database mysql php web

mysql
Install and configure mysql on your system.
robertdebock build passing 4.8 / 5 Score 14762 Downloads Last Imported: 5 days ago
Tags: alpine centos database debian fedora installer mariadb mysql package rhel ubuntu

mysql
MySQL server for RHEL/CentOS and Debian/Ubuntu.
unxnn build passing 5 / 5 Score 23304 Downloads Last Imported: 4 months ago
Tags: database db mariadb mysql sql

\$ ansible-galaxy search mysql

Found 1126 roles matching your search. Showing first 1000.

Name	Description
Outsider.ansible_zabbix_agent	Installing and maintaining zabbix-agent for install and configure unattended upgrade
1mr.unattended	Simply installs MySQL 5.7 on Xenial.
1nfinitum.mysql	Instalacao e Configuracao do servidor MySQL
4linuxdevops.mysql-server	Install and configure MySQL Database
5KYDEVOP5.skydevops-mysql	Manage Yourls, a URL shortener web app.
AAbouZaid.yourls	your description
AAROC.AAROC_fg-db	Simple deployment tool with hooks
aaronpederson.ansible-autodeploy	Install and configure mysqld_exporter
abednarik.mysqld-exporter	OpenStack Neutron controller node
abelboldu.openstack-glance	OpenStack Nova controller node
abelboldu.openstack-keystone	configure mysql-backup with xtrabackup and
abelboldu.openstack-neutron-controller	Install mysql-server package
abelboldu.openstack-nova-controller	achaussier.mysql-backup
achaussier.mysql-server	achilleskal.ansible_mysql8
achilleskal.ansible_mysql8	your description
adarnimrod.mysql	Provision a MySQL server

Use Role

```
$ ansible-galaxy install geerlingguy.mysql
```

```
- downloading role 'mysql', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-mysql/archive/2.9.5.tar.gz
- extracting geerlingguy.mysql to /etc/ansible/roles/etc/ansible/roles/geerlingguy.mysql
- geerlingguy.mysql (2.9.5) was installed successfully
```

```
playbook.yml
```

```
-  
  name: Install and Configure MySQL  
  hosts: db-server  
  roles:  
    - geerlingguy.mysql  
  
-  
  name: Install and Configure MySQL  
  hosts: db-server  
  roles:  
    - role: geerlingguy.mysql  
      become: yes  
    vars:  
      mysql_user_name: db-user
```

Use Role

Playbook-all-in-one.yml

```
-  
  name: Install and Configure MySQL  
  hosts: db-and-webserver  
  roles:  
    - geerlingguy.mysql  
    - nginx
```



Playbook-distributed.yml

```
-  
  name: Install and Configure MySQL  
  hosts: db-server  
  roles:  
    - geerlingguy.mysql  
  
-  
  name: Install and Configure Web Server  
  hosts: web-server  
  roles:  
    - nginx
```



List Roles

```
$ ansible-galaxy list
```

- geerlingguy.mysql
- kodekloud1.mysql

```
$ ansible-config dump | grep ROLE
```

```
DEFAULT_PRIVATE_ROLE_VARS(default) = False
DEFAULT_ROLES_PATH(default) = [u'/root/.ansible/roles', u'/usr/share/ansible/roles', u'/etc/ansible/roles']
GALAXY_ROLE_SKELETON(default) = None
GALAXY_ROLE_SKELETON_IGNORE(default) = ['^\.git$', '^.*\.git_keep$']
```

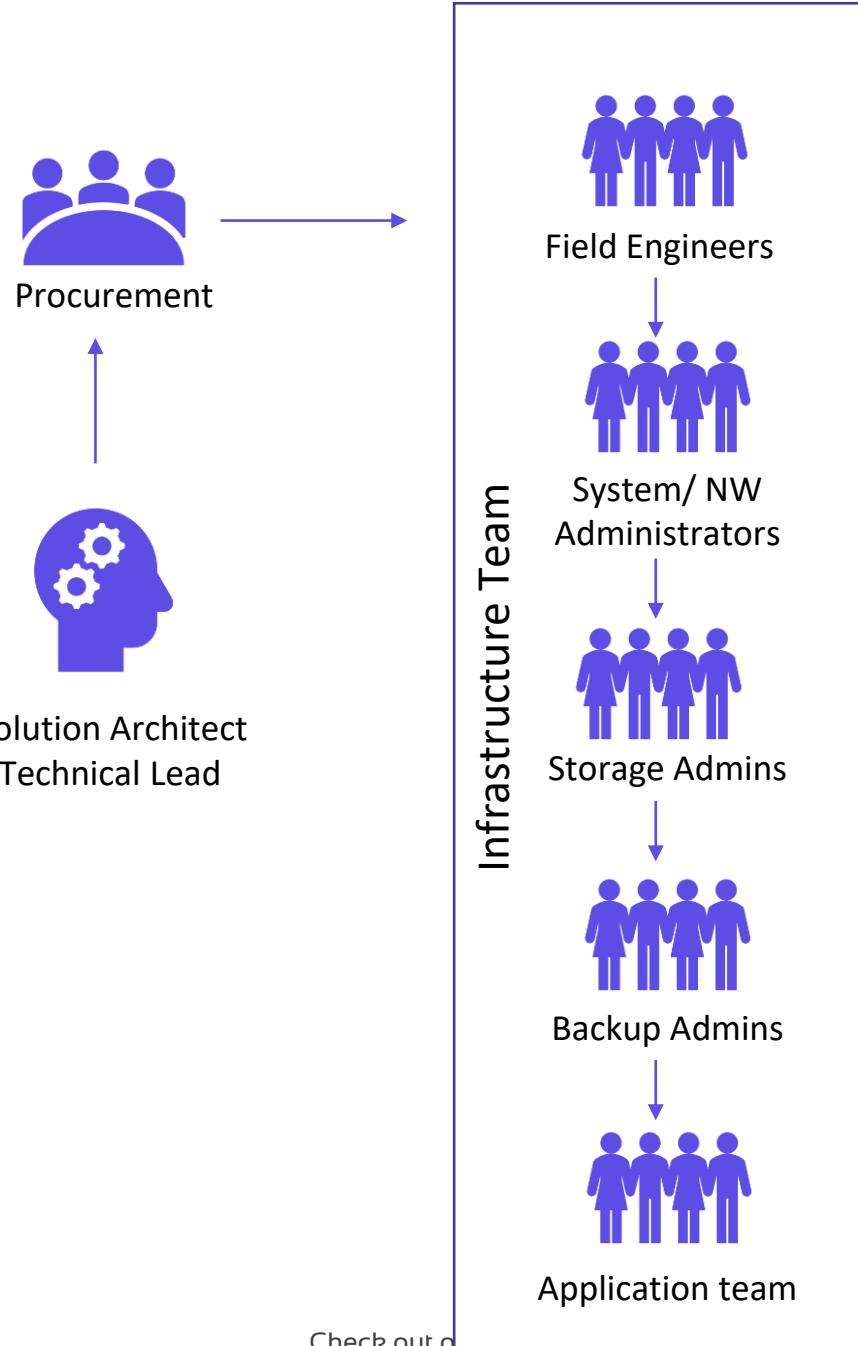
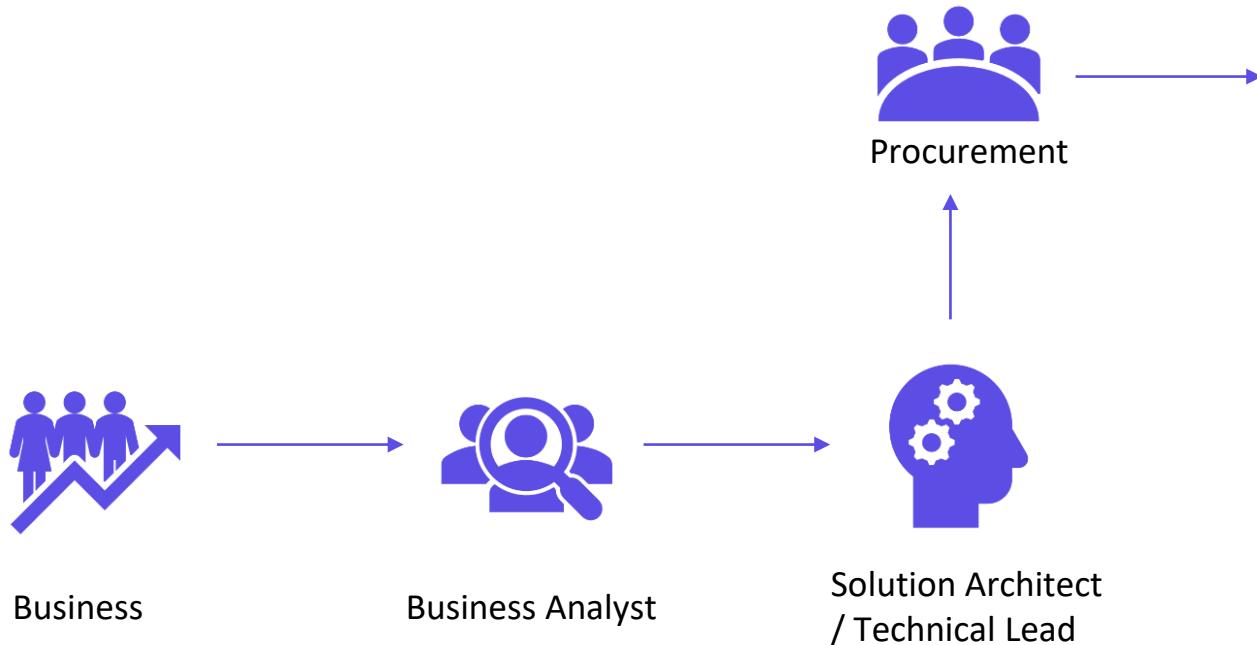
```
$ ansible-galaxy install geerlingguy.mysql -p ./roles
```

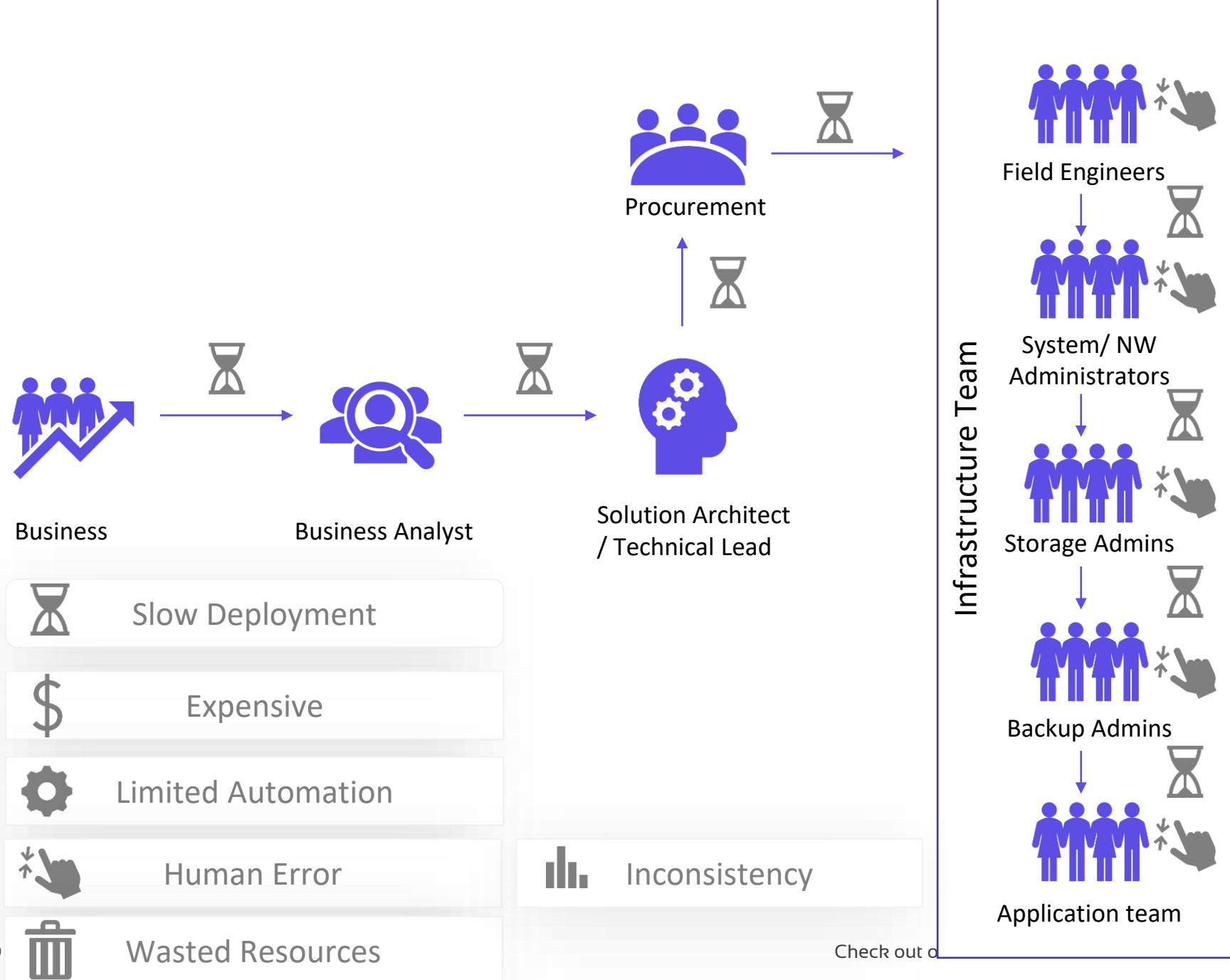


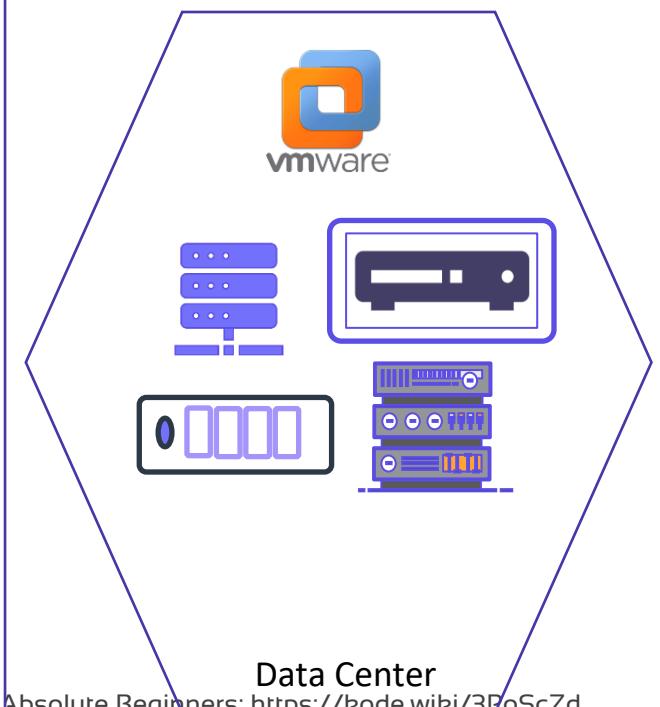
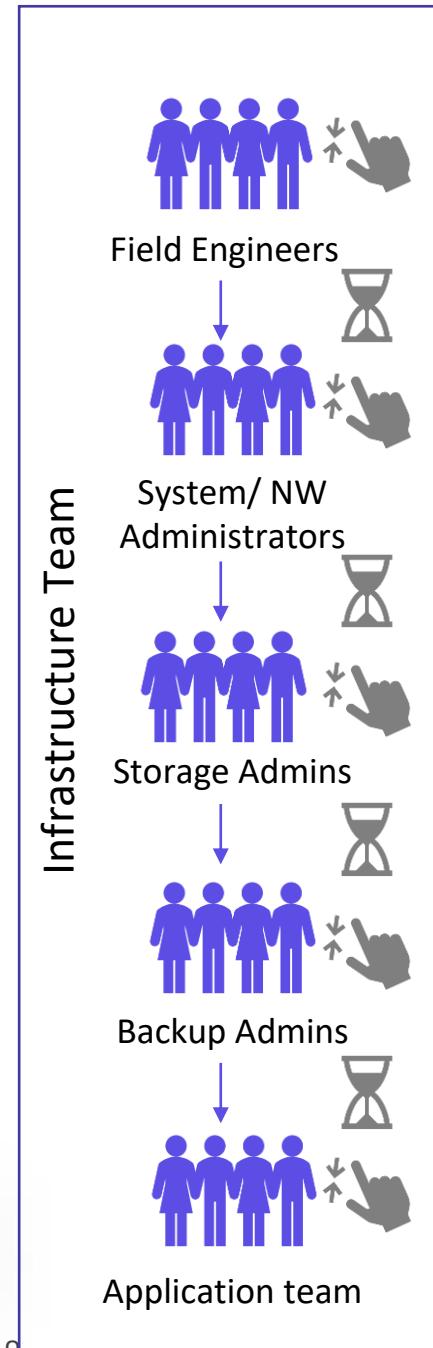
KodeKloud

Check out our full course on Ansible for the Absolute Beginners: <https://kode.wiki/3No1wef>

Traditional IT & Challenges







Check out our

Absolute Beginners: <https://kode.wiki/3D0ScZd>



Services ▾

Resource Groups ▾



1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

**Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0b1e2eeb33ce3d66f**

Free tier
eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extra

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name **launch-wizard-1**

Description **launch-wizard-1 created 2020-07-09T15:48:36.426-04:00**

Type (i)	Protocol (i)	Port Range (i)	Source (i)	Description (i)
---	---	---	---	--

This security group has no rules

Instance Details

Number of instances **1**

Network **vpc-fe3baa86**

Subnet **No preference (default subnet in any Availability Zone)**

Purchasing option **On demand**

Shell

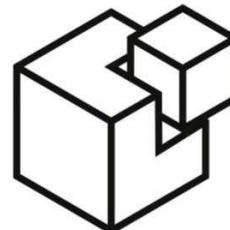
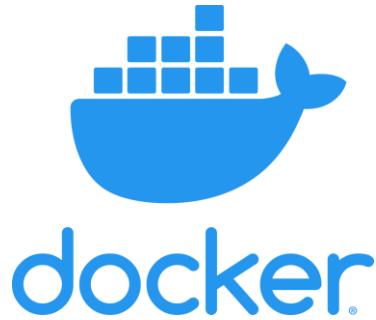
Python

Ruby

Perl

Powershell

Infrastructure as Code



SALTSTACK





KodeKloud

Check out our full course on Terraform for the Absolute Beginners: <https://kode.wiki/3PoScZd>

Infrastructure as Code

Infrastructure as Code

ec2.sh

```
#!/bin/bash

IP_ADDRESS="10.2.2.1"

EC2_INSTANCE=$(ec2-run-instances --instance-type t2.micro ami-0edab43b6fa892279)

INSTANCE=$(echo ${EC2_INSTANCE} | sed 's/*INSTANCE //'
| sed 's/ .*//')

# Wait for instance to be ready
while ! ec2-describe-instances $INSTANCE | grep -q "running"
do
    echo Waiting for $INSTANCE is to be ready...
done

# Check if instance is not provisioned and exit
if [ ! $(ec2-describe-instances $INSTANCE | grep -q "running") ]; then
    echo Instance $INSTANCE is stopped.
    exit
fi

ec2-associate-address $IP_ADDRESS -i $INSTANCE

echo Instance $INSTANCE was created successfully!!!
```

The screenshot shows the AWS Step 7: Review Instance Launch wizard. The top navigation bar includes the AWS logo, Services dropdown, Resource Groups dropdown, and a star icon. Below the navigation, a horizontal menu bar lists steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, and 6. Configure Security Groups. The current step, Step 7, is highlighted.

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign security groups and launch the instance.

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0b1e2eeb33ce3d66f
Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance.

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)
t2.micro	Variable	1	1	EBS only

Security Groups

Security group name	Description
launch-wizard-1	launch-wizard-1 created 2020-07-09T15:48:36.426-04:00

This security group has

Instance Details

Number of instances	1
Network	vpc-fe3baa86

Infrastructure as Code

ec2.sh

```
#!/bin/bash

IP_ADDRESS="10.2.2.1"

EC2_INSTANCE=$(ec2-run-instances --instance-type t2.micro ami-0edab43b6fa892279)

INSTANCE=$(echo ${EC2_INSTANCE} | sed 's/*INSTANCE //'
| sed 's/ .*//')

# Wait for instance to be ready
while ! ec2-describe-instances $INSTANCE | grep -q "running"
do
    echo Waiting for $INSTANCE is to be ready...
done

# Check if instance is not provisioned and exit
if [ ! $(ec2-describe-instances $INSTANCE | grep -q "running") ]; then
    echo Instance $INSTANCE is stopped.
    exit
fi

ec2-associate-address $IP_ADDRESS -i $INSTANCE

echo Instance $INSTANCE was created successfully!!!
```

main.tf

```
resource "aws_instance" "webserver" {
    ami           = "ami-0edab43b6fa892279"
    instance_type = "t2.micro"
}
```

Infrastructure as Code

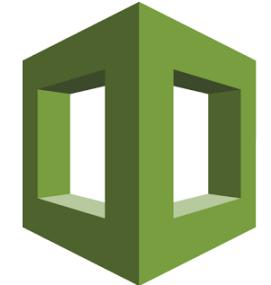
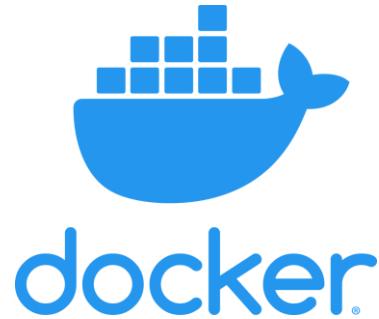
ec2.yaml

```
- amazon.aws.ec2:  
    key_name: mykey  
    instance_type: t2.micro  
    image: ami-123456  
    wait: yes  
    group: webserver  
    count: 3  
    vpc_subnet_id: subnet-29e63245  
    assign_public_ip: yes
```

main.tf

```
resource "aws_instance" "webserver" {  
    ami           = "ami-0edab43b6fa892279"  
    instance_type = "t2.micro"  
}
```

Types of IAC Tools



SALTSTACK



Types of IAC Tools

Configuration Management



Server Templating



HashiCorp
Vagrant

Provisioning Tools



Types of IAC Tools

Configuration Management



ANSIBLE



Designed to Install and Manage Software

Maintains Standard Structure

Version Control

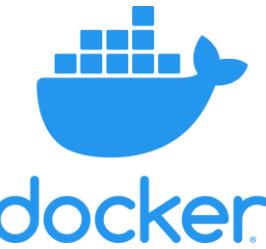
Idempotent

Server Templating Tools

Pre Installed Software and Dependencies

Virtual Machine or Docker Images

Immutable Infrastructure



Provisioning Tools

Deploy Immutable Infrastructure resources

Servers, Databases, Network Components etc.

Multiple Providers



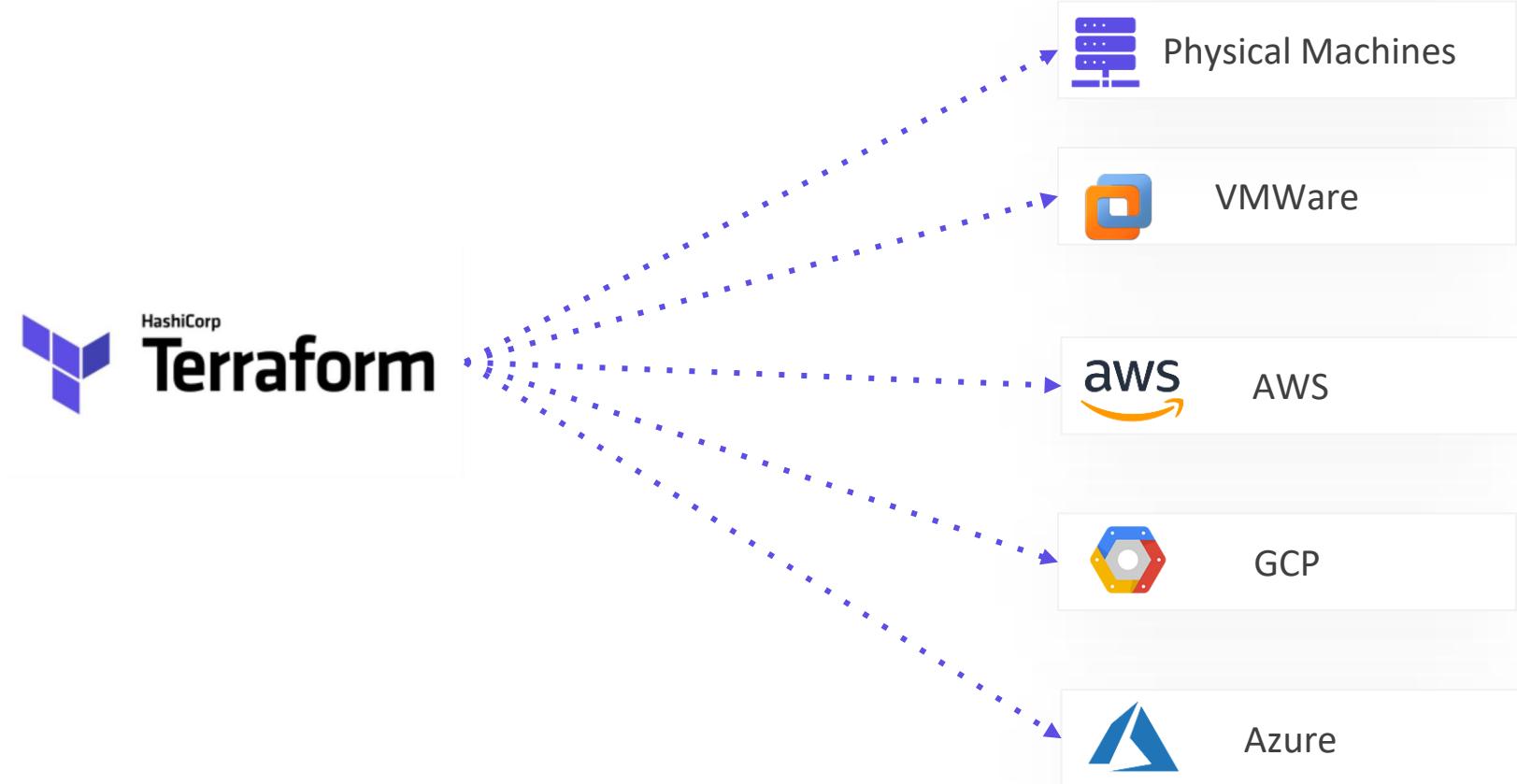


KodeKloud

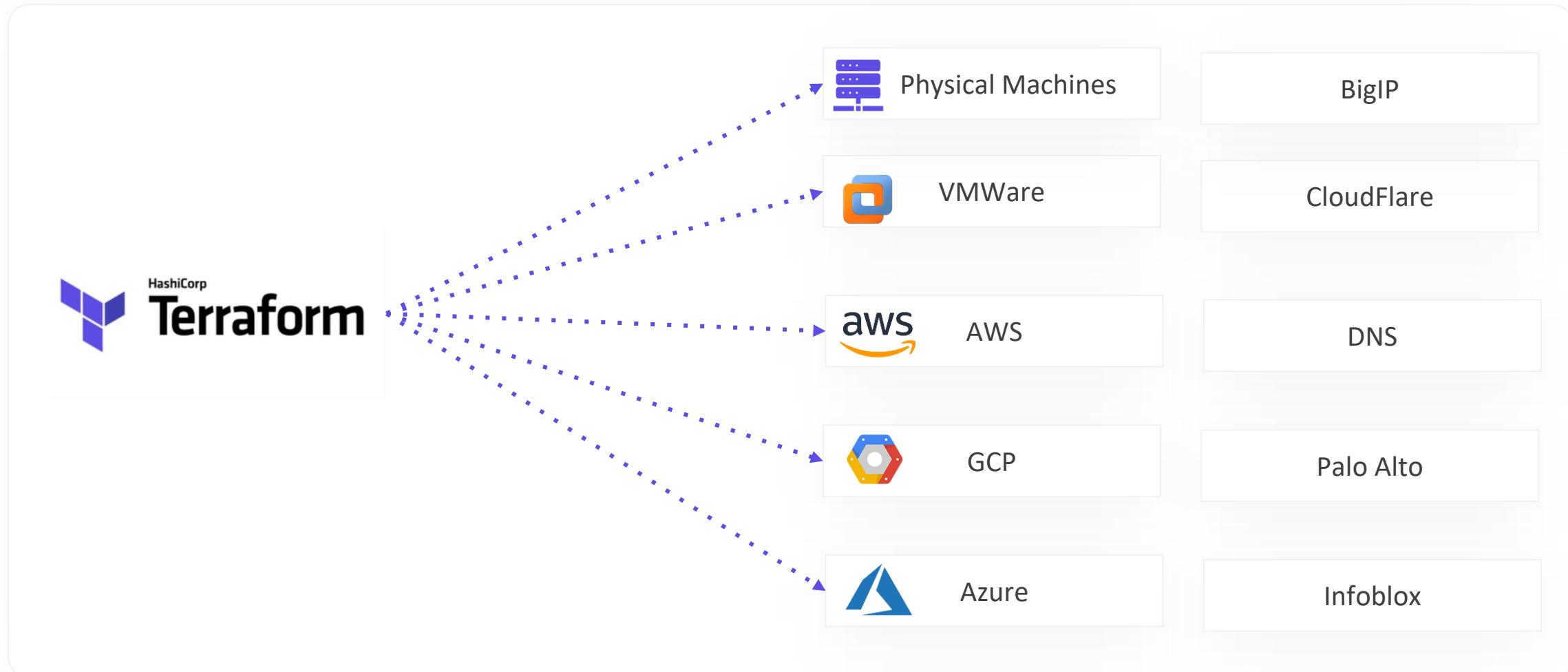
Check out our full course on Terraform for the Absolute Beginners: <https://kode.wiki/3PoScZd>

Terraform

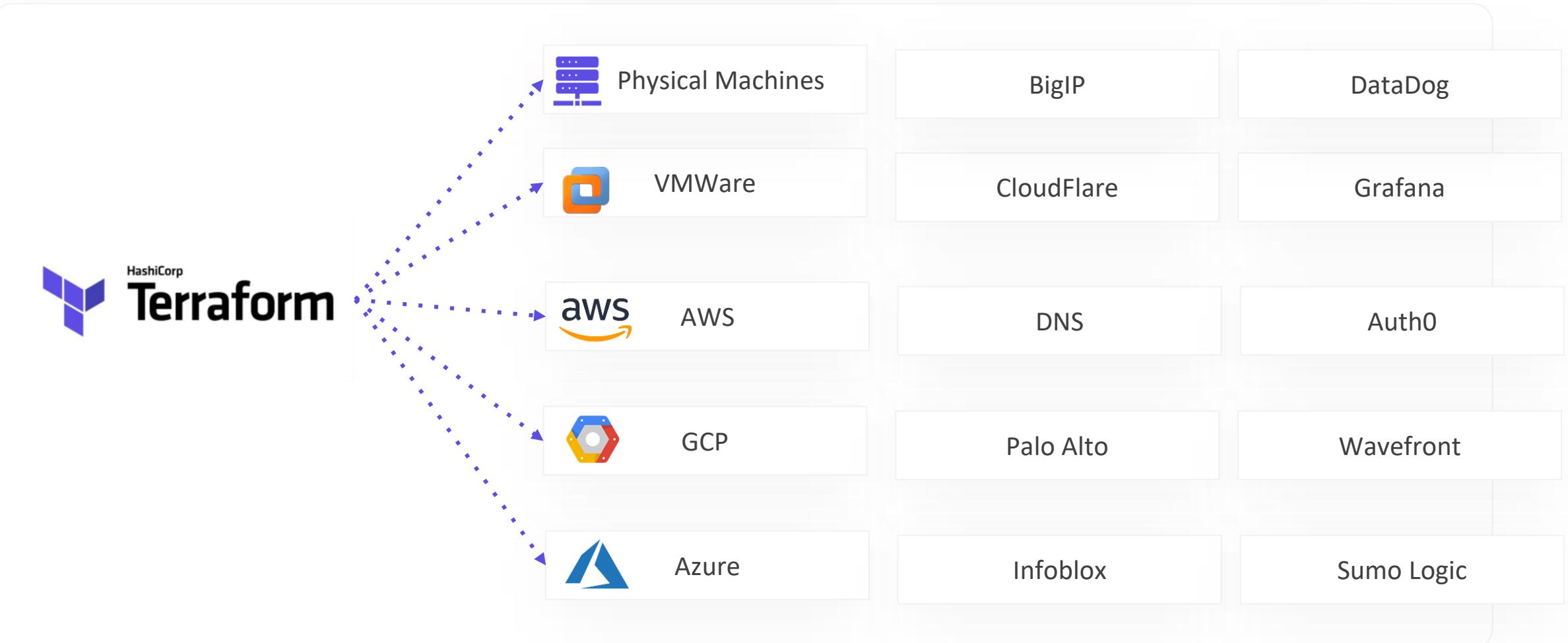
Why Terraform?



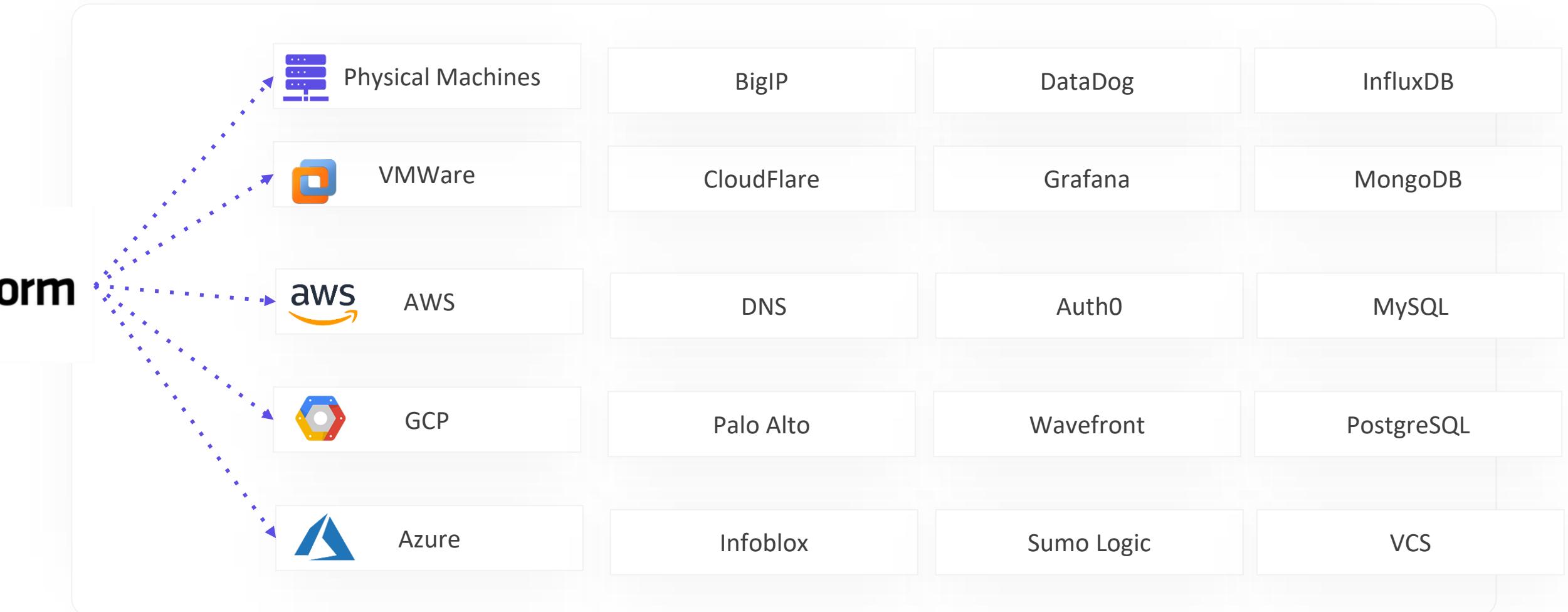
Providers



Providers



Providers



HashiCorp Configuration Language

```
main.tf

resource "aws_instance" "webserver" {
    ami           = "ami-0edab43b6fa892279"
    instance_type = "t2.micro"
}

resource "aws_s3_bucket" "finance" {
    bucket = "finanace-21092020"
    tags   = {
        Description = "Finance and Payroll"
    }
}

resource "aws_iam_user" "admin-user" {
    name = "lucy"
    tags = {
        Description = "Team Leader"
    }
}
```

Declarative

main.tf

```
resource "aws_instance" "webserver" {
    ami           = "ami-0edab43b6fa892279"
    instance_type = "t2.micro"
}

resource "aws_s3_bucket" "finance" {
    bucket = "finanace-21092020"
    tags   = {
        Description = "Finance and Payroll"
    }
}

resource "aws_iam_user" "admin-user" {
    name = "lucy"
    tags = {
        Description = "Team Leader"
    }
}
```

Real World Infrastructure

Declarative

main.tf

```
resource "aws_instance" "webserver" {  
    ami           = "ami-0edab43b6fa892279"  
    instance_type = "t2.micro"  
}  
  
resource "aws_s3_bucket" "finance" {  
    bucket = "finanace-21092020"  
    tags   = {  
        Description = "Finance and Payroll"  
    }  
}  
  
resource "aws_iam_user" "admin-user" {  
    name = "lucy"  
    tags = {  
        Description = "Team Leader"  
    }  
}
```

Init

Plan

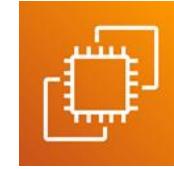
Apply

Real World Infrastructure

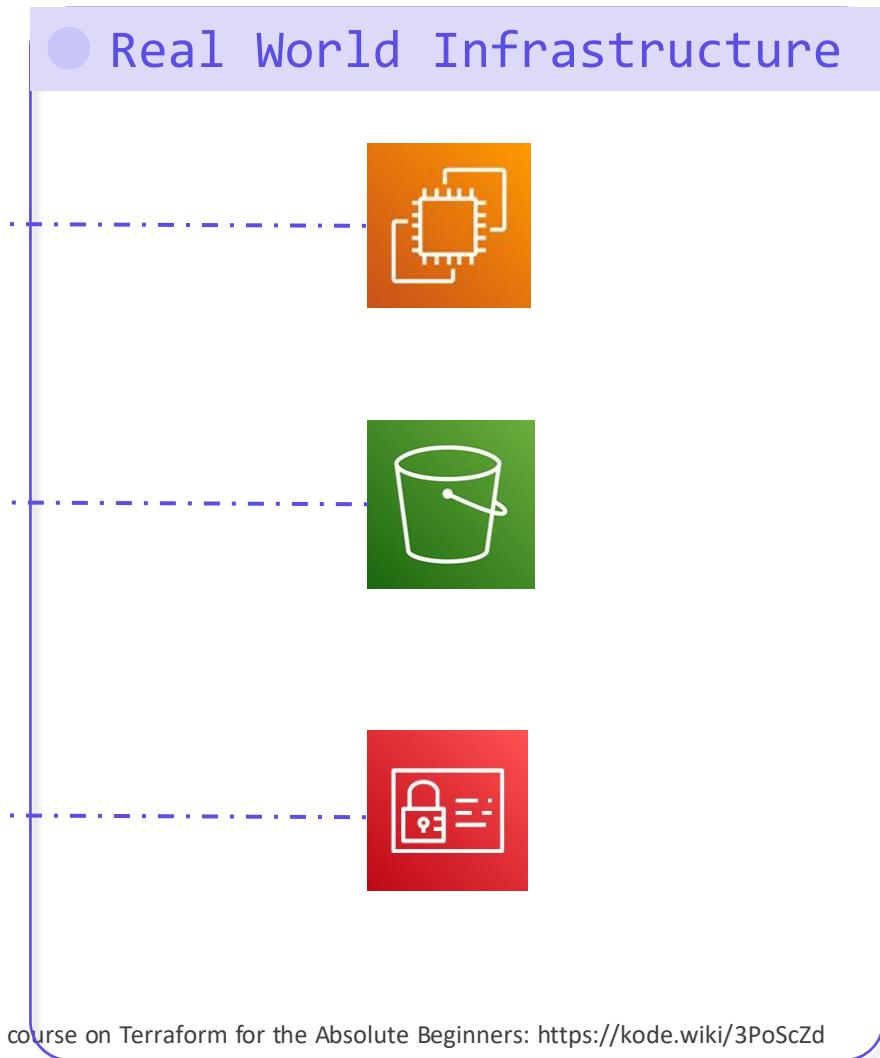
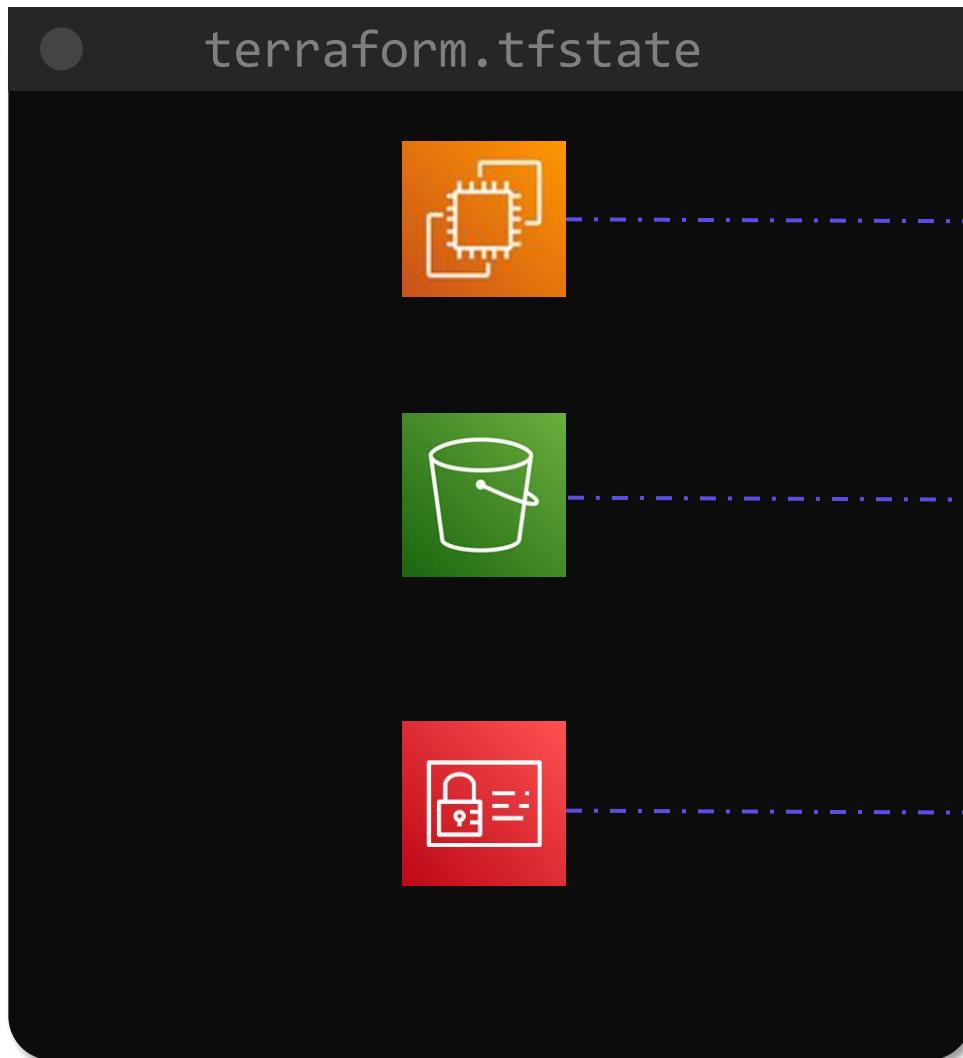


Resource

Real World Infrastructure

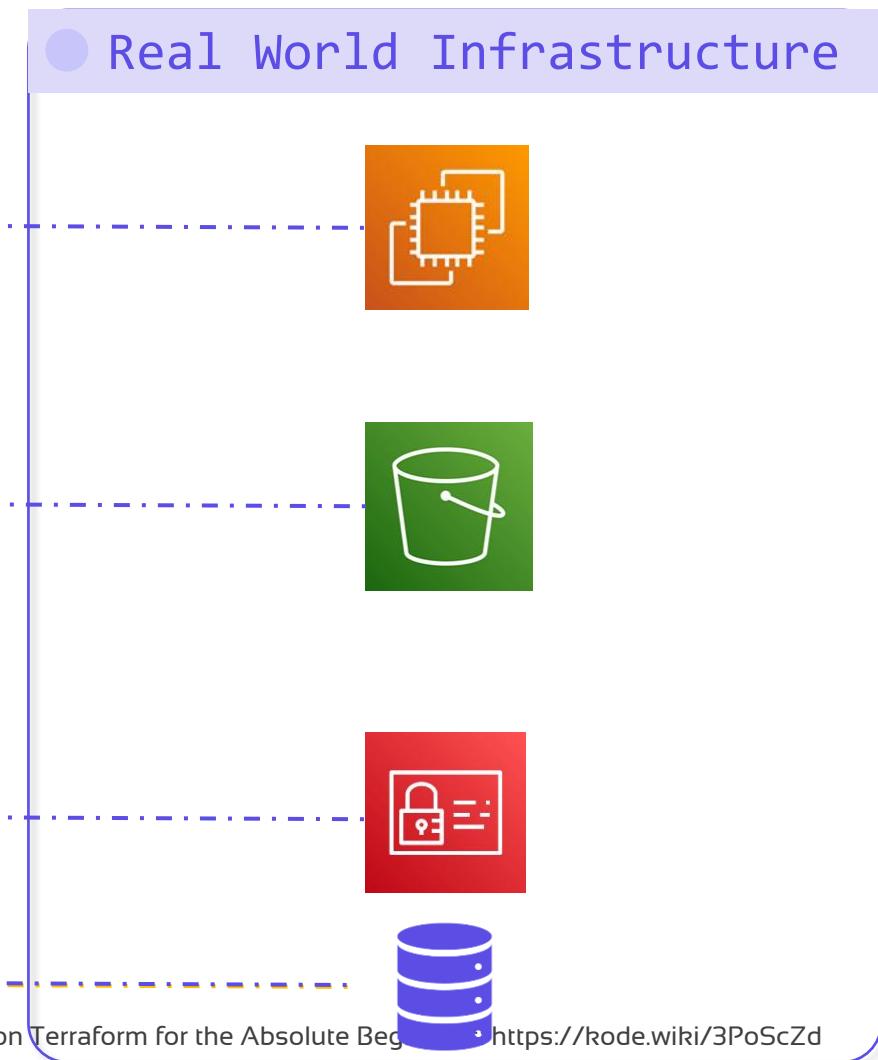
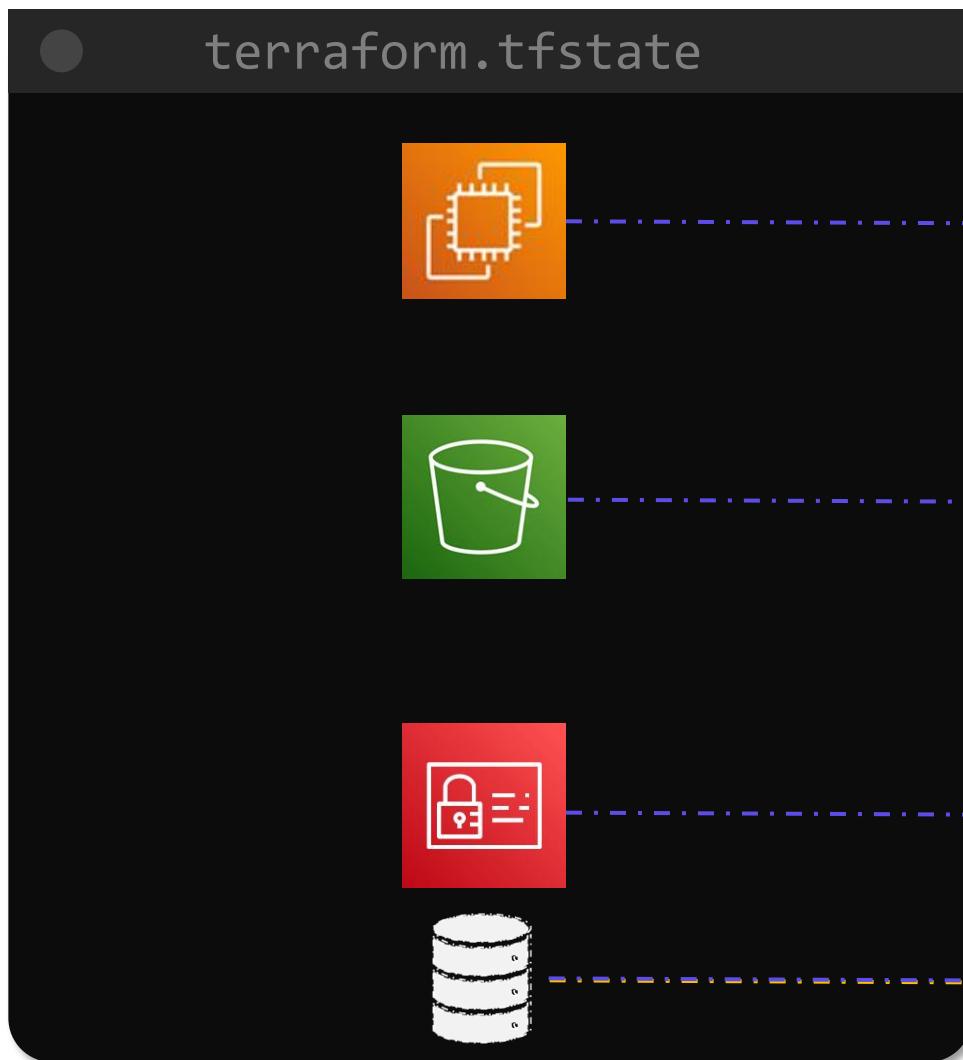


Terraform State



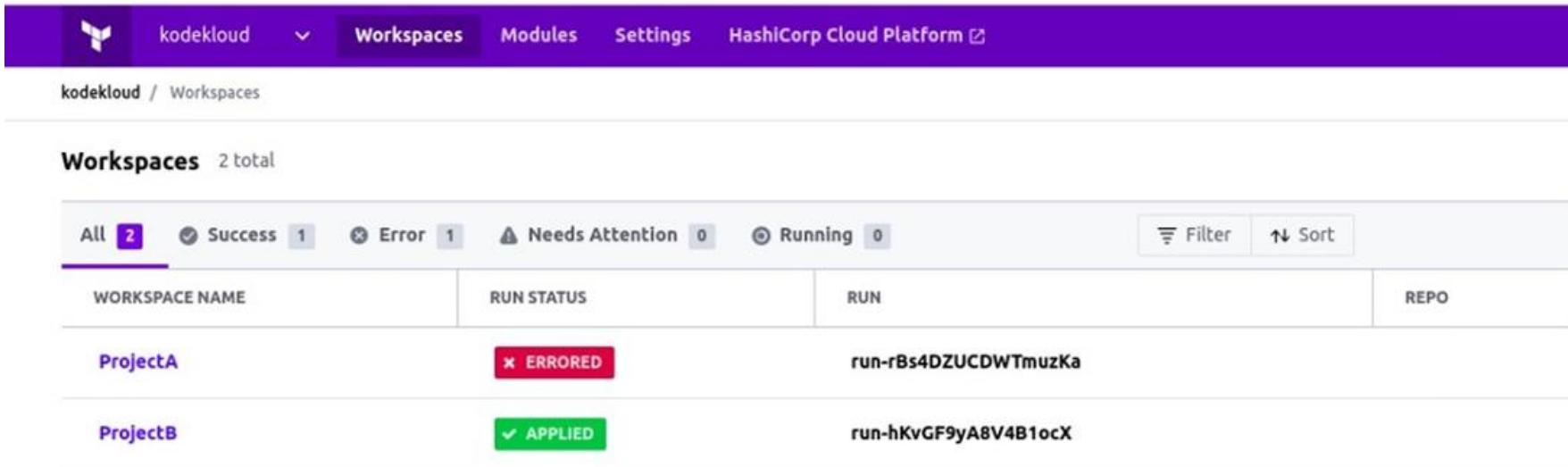
Check out our full course on Terraform for the Absolute Beginners: <https://kode.wiki/3PoScZd>

Terraform Import



Check out our full course on Terraform for the Absolute Beginner: <https://kode.wiki/3PoScZd>

Terraform Cloud and Terraform Enterprise



The screenshot shows the Terraform Cloud Workspaces interface. At the top, there's a purple header bar with the KodeKloud logo, the account name "kodekloud", and navigation links for "Workspaces", "Modules", "Settings", and "HashiCorp Cloud Platform". Below the header, the URL "kodekloud / Workspaces" is displayed. The main area is titled "Workspaces" with "2 total". A filter bar at the top of the table allows selecting "All" (selected), "Success", "Error", "Needs Attention", or "Running". It also includes "Filter" and "Sort" buttons. The table has columns for "WORKSPACE NAME", "RUN STATUS", "RUN", and "REPO". The first row, "ProjectA", has a red "X" icon in the status column, indicating an error. The second row, "ProjectB", has a green checkmark icon, indicating it is applied. The "RUN" column shows the run ID for each project.

WORKSPACE NAME	RUN STATUS	RUN	REPO
ProjectA	✗ ERRORED	run-rBs4DZUCDWtMuzKa	
ProjectB	✓ APPLIED	run-hKvGF9yA8V4B1ocX	



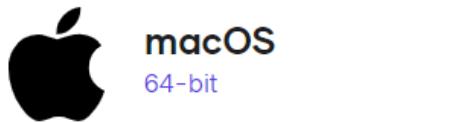
KodeKloud

Check out our full course on Terraform for the Absolute Beginners: <https://kode.wiki/3PoScZd>

Installing Terraform

>_

```
$ wget https://releases.hashicorp.com/terraform/0.13.0/terraform_0.13.0_linux_amd64.zip  
$ unzip terraform_0.13.0_linux_amd64.zip  
$ mv terraform /usr/local/bin  
$ terraform version  
Terraform v0.13.0
```



macOS

64-bit



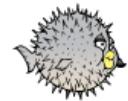
FreeBSD

32-bit | 64-bit | Arm



Linux

32-bit | 64-bit | Arm



OpenBSD

32-bit | 64-bit



Solaris

64-bit



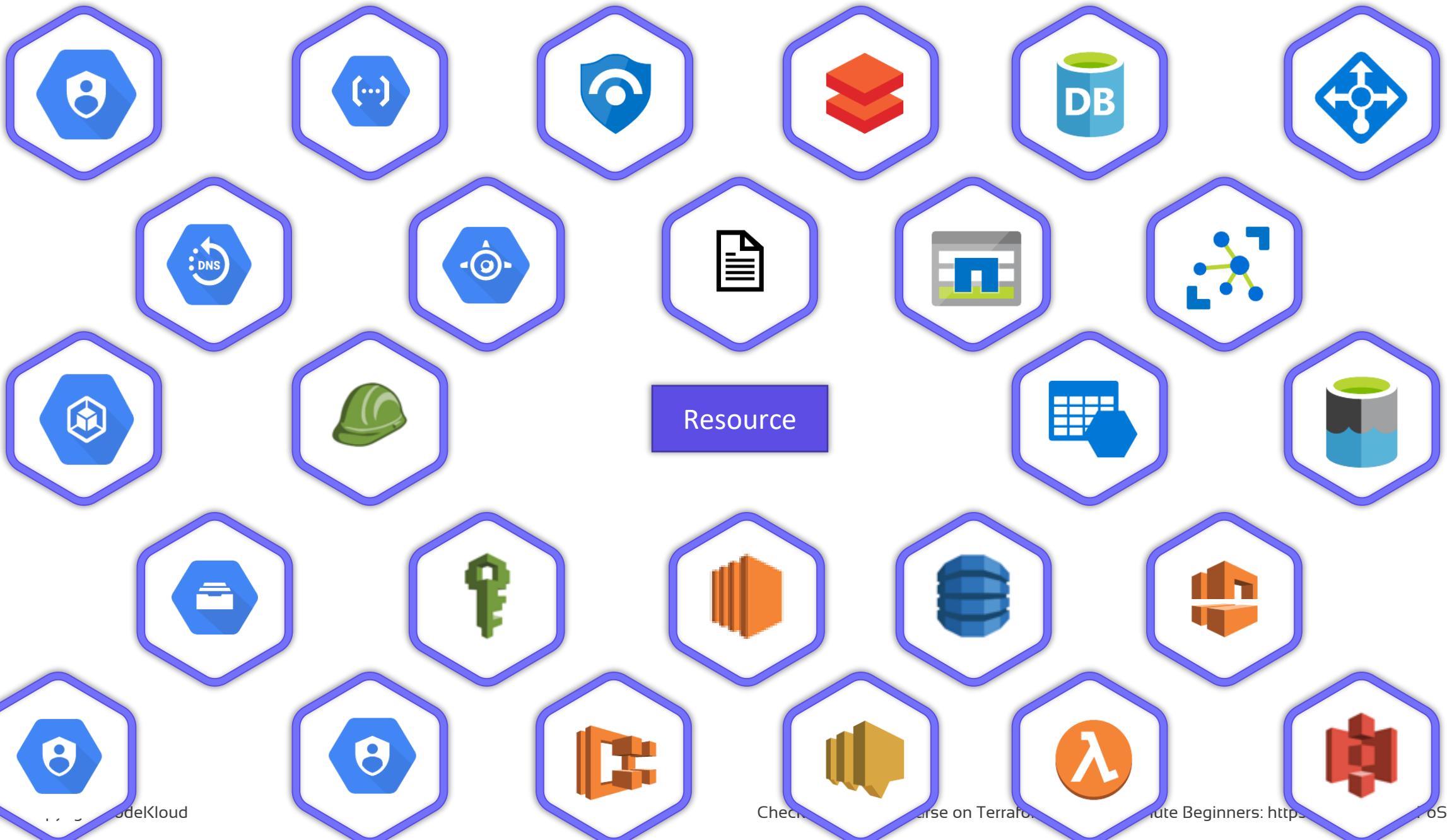
Windows

32-bit | 64-bit

HCL – Declarative Language

```
aws.tf
```

```
resource "aws_instance" "webserver" {
    ami = "ami-0c2f25c1f66a1ff4d"
    instance_type = "t2.micro"
}
```



nodeKloud

Check out our course on Terraform for Absolute Beginners: <https://www.udemy.com/course/terraform-for-absolute-beginners/>



Resource



HCL Basics

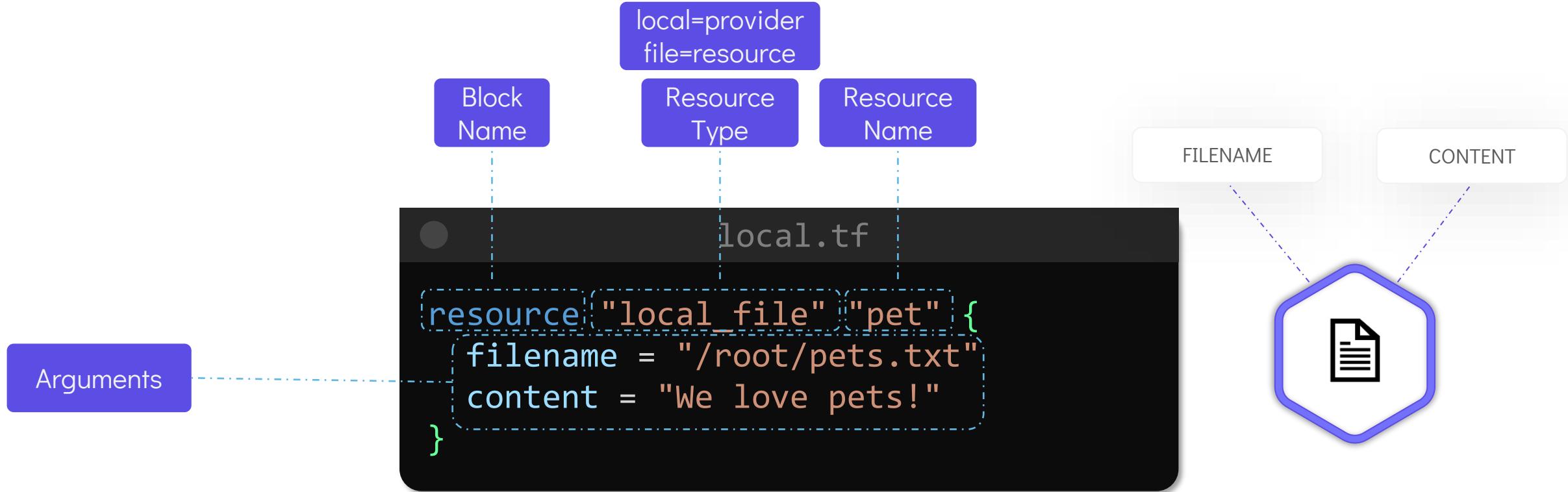
```
>_
```

```
$ mkdir /root/terraform-local-file  
$ cd /root/terraform-local-file
```

```
local.tf
```

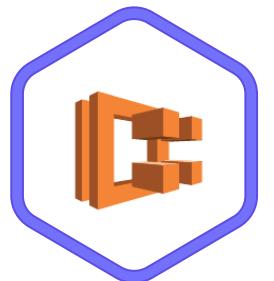
```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
}
```





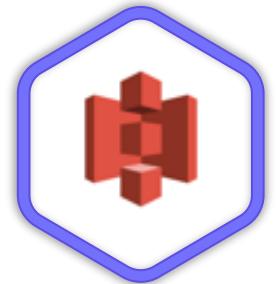
```
aws-ec2.tf
```

```
resource "aws_instance" "webserver" {
    ami = "ami-0c2f25c1f66a1ff4d"
    instance_type = "t2.micro"
}
```



aws-s3.tf

```
resource "aws_s3_bucket" "data" {
    bucket = "webserver-bucket-org-2207"
    acl    = "private"
}
```



local.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
}
```



local.tf

```
resource "local_file" "pet" {
  filename = "/root/pets.txt"
  content = "We love pets!"
}
```



```
>_
```

```
$ terraform init
```

```
Initializing the backend...
```

```
Initializing provider plugins...
```

```
- Finding latest version of hashicorp/local...
```

```
- Installing hashicorp/local v1.4.0...
```

```
- Installed hashicorp/local v1.4.0 (signed by HashiCorp)
```

```
The following providers do not have any version constraints in configuration,  
so the latest version was installed.
```

```
To prevent automatic upgrades to new major versions that may contain breaking  
changes, we recommend adding version constraints in a required_providers block  
in your configuration, with the constraint strings suggested below.
```

```
* hashicorp/local: version = "~> 1.4.0"
```

```
Terraform has been successfully initialized!
```



>_

```
$ terraform plan
```

```
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.
```

```
-----
```

An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

```
[+ create]
```

Terraform will perform the following actions:

```
# local_file.pet will be created
+ resource "local_file" "pet" {
    + content          = "We love pets!"
    + directory_permission = "0777"
    + file_permission      = "0777"
    + filename           = "/root/pets.txt"
    + id                 = (known after apply)
}
```

```
Plan: 1 to add, 0 to change, 0 to destroy.
```

```
-----
```

Note: You didn't specify an "-out" parameter to save this plan, so
Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.



>_

```
$ terraform apply
```

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

```
[+] # local_file.pet will be created!
+ resource "local_file" "pet" {}
  + content          = "We love pets!"
  + directory_permission = "0777"
  + file_permission    = "0777"
  + filename           = "/root/pets.txt"
  + id                 = (known after apply)
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?

Terraform will perform the actions described above.

Only 'yes' will be accepted to approve.

```
[+] Enter a value: yes
local_file.new_file: Creating...
local_file.new_file: Creation complete after 0s
[id=521c5c732c78cb42cc9513ecc7c0638c4a115b55]
[Apply complete! Resources: 1 added, 0 changed, 0 destroyed.]
```

```
$ cat /root/pets.txt
```

We love pets!

>_

```
$ terraform show  
# local_file.pet:  
resource "local_file" "pet" {  
    content          = "We love pets!"  
    directory_permission = "0777"  
    file_permission     = "0777"  
    filename           = "/root/pets.txt"  
    id                 = "cba595b7d9f94ba1107a46f3f731912d95fb3d2c"  
}
```



local=provider
file=resource
Resource
Type

local.tf

```
resource("local_file") "pet" {
  filename = "/root/pets.txt"
  content = "We love pets!"
}
```



provider



resource_type



Arguments

Argument-1

Argument-1

Argument-1

Argument-2

Argument-2

Argument-2

Argument-X

Argument-X

Argument-X

Argument-1

Argument-2

Argument-X

Argument-1

Argument-2

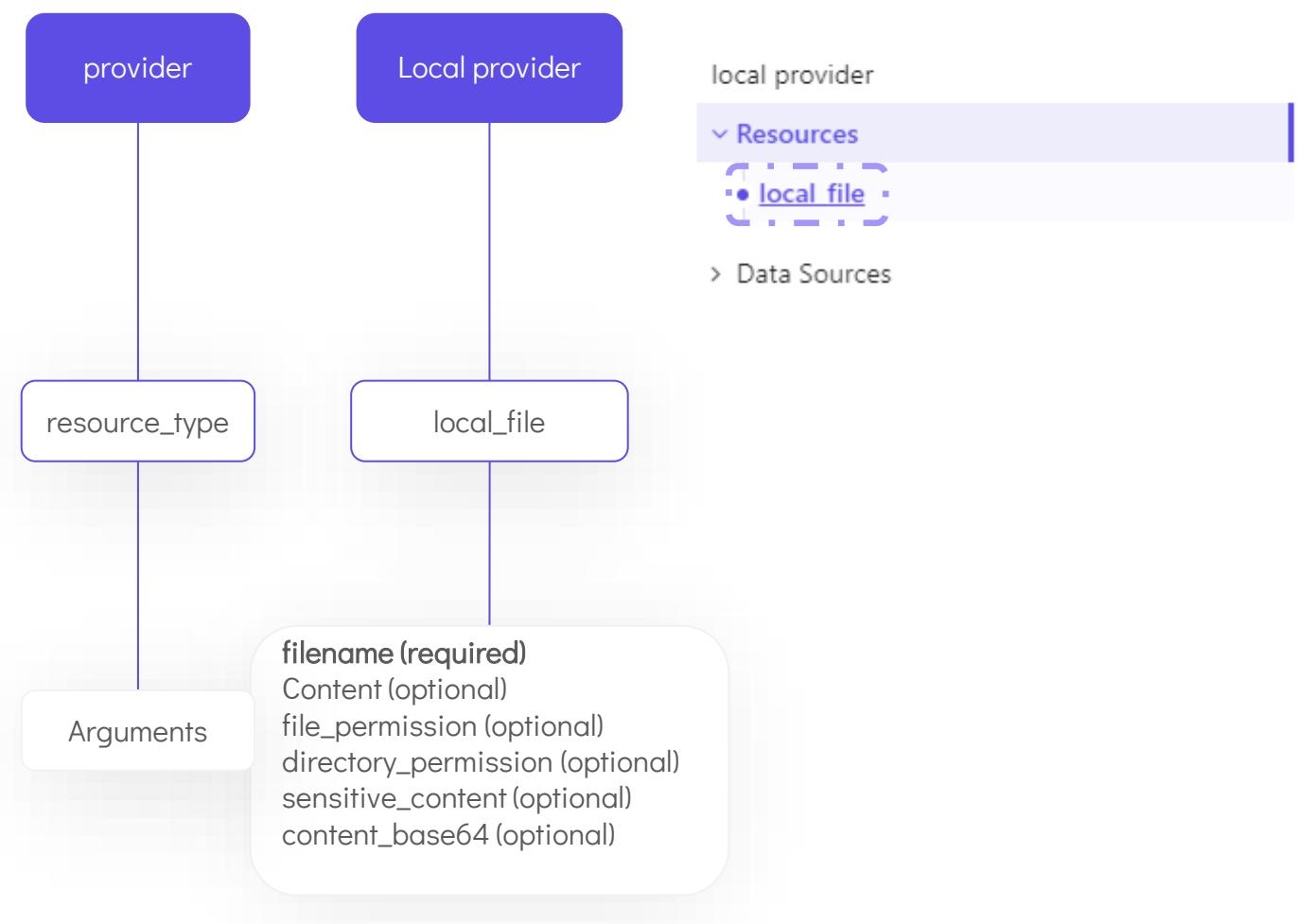
Argument-X

Argument-1

Argument-2

Argument-X



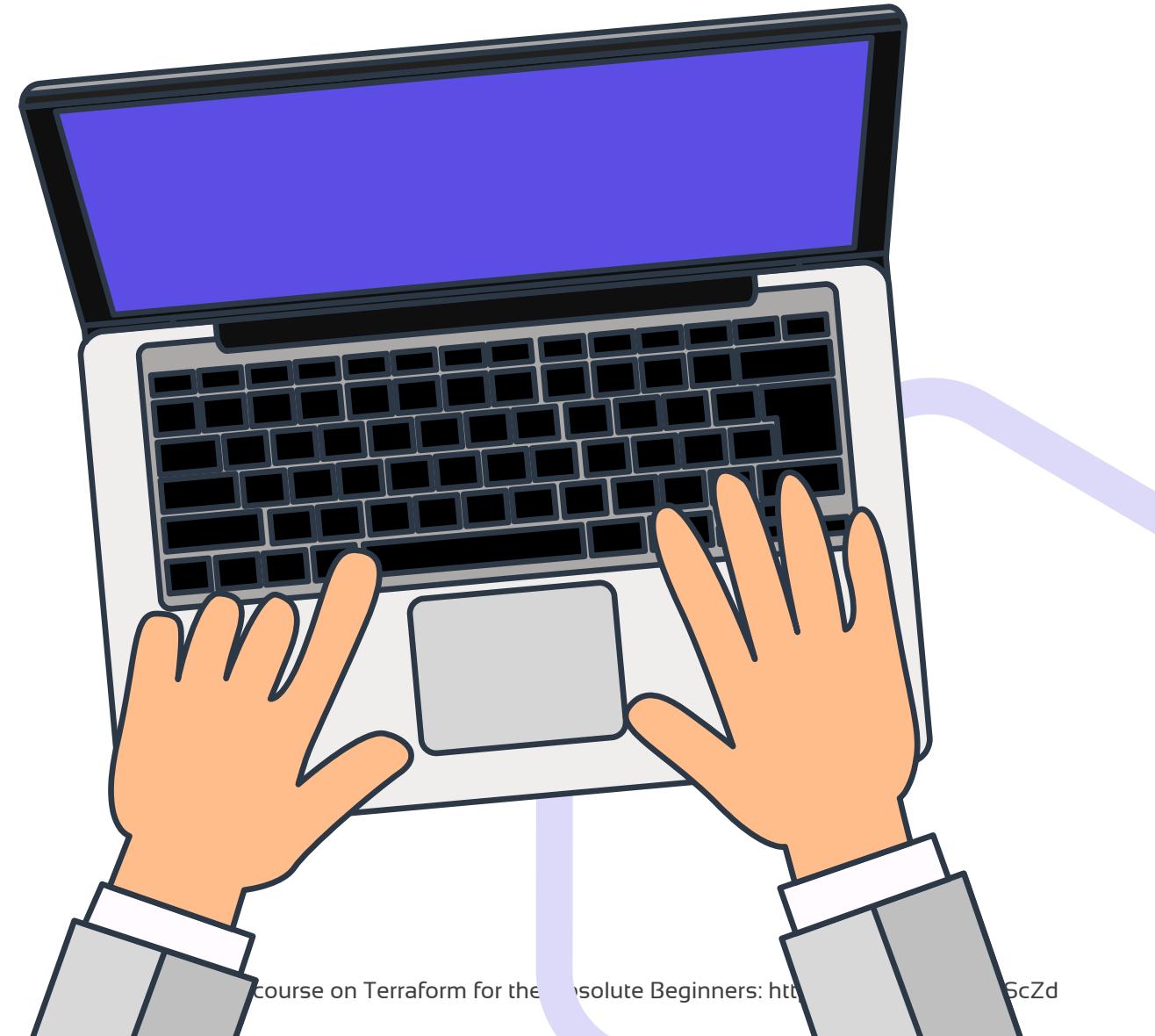


Argument Reference

The following arguments are supported:

- `content` - (Optional) The content of file to create. Conflicts with `sensitive_content` .
- `sensitive_content` - (Optional) The content of file to create. Will not be encoded. Conflicts with `content` and `content_base64` .
- `content_base64` - (Optional) The base64 encoded content of the file to create when dealing with binary data. Conflicts with `content` and `sensitive_content` .
- `filename` - (Required) The path of the file to create.
- `file_permission` - (Optional) The permission to set for the created file. Expects a string. The default value is `"0777"` .
- `directory_permission` - (Optional) The permission to set for any directory in the path. Expects a string. The default value is `"0777"` .

HANDS-ON LABS



Update and Destroy Infrastructure

local.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
    file_permission = "0700"  
}
```



```
$ terraform plan
```

```
local_file.pet: Refreshing state...  
[id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]
```

```
-----  
An execution plan has been generated and is shown below.  
Resource actions are indicated with the following symbols:  
-/+ destroy and then create replacement
```

```
Terraform will perform the following actions:
```

```
[# local_file.pet must be replaced]  
-/+ resource "local_file" "pet" {  
    content          = "We love pets!"  
    directory_permission = "0777"  
    file_permission   = "0777" -> "0700" # forces replacement  
    filename         = "/root/pet.txt"  
    ~ id             =  
"5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf" -> (known after apply)  
}  
Plan: 1 to add, 0 to change, 1 to destroy.  
-----
```

```
Note: You didn't specify an "-out" parameter to save this plan, so  
Terraform  
can't guarantee that exactly these actions will be performed if  
"terraform apply" is subsequently run.
```

>_

```
$ ls -ltr /root/pets.txt  
-rwx----- 1 root root 30 Aug 17 23:20 pet.txt
```



>_

```
$ terraform apply
```

```
# local_file.pet must be replaced  
-/+ resource "local_file" "pet" {  
    content          = "We love pets!"  
    directory_permission = "0777"  
    ~ file_permission      = "0777" -> "0700" # forces replacement  
    filename         = "/root/pet.txt"  
    ~ id              =  
    "5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf" -> (known after apply)  
}
```

```
Plan: 1 to add, 0 to change, 1 to destroy.
```

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

```
local_file.pet: Destroying...  
[id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]  
local_file.pet: Destruction complete after 0s  
local_file.pet: Creating...  
local_file.pet: Creation complete after 0s  
[id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]
```

```
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
```



```
>_
$ terraform destroy
local_file.pet: Refreshing state...
[ id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

  # local_file.pet.will.be.destroyed
  - resource "local_file" "pet" {
      content          = "My favorite pet is a gold fish" -> null
      directory_permission = "0777" -> null
      file_permission    = "0700" -> null
      filename          = "/root/pet.txt" -> null
      id                = "5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf" -
    > null
    }

Plan: 0 to add, 0 to change, 1 to destroy.

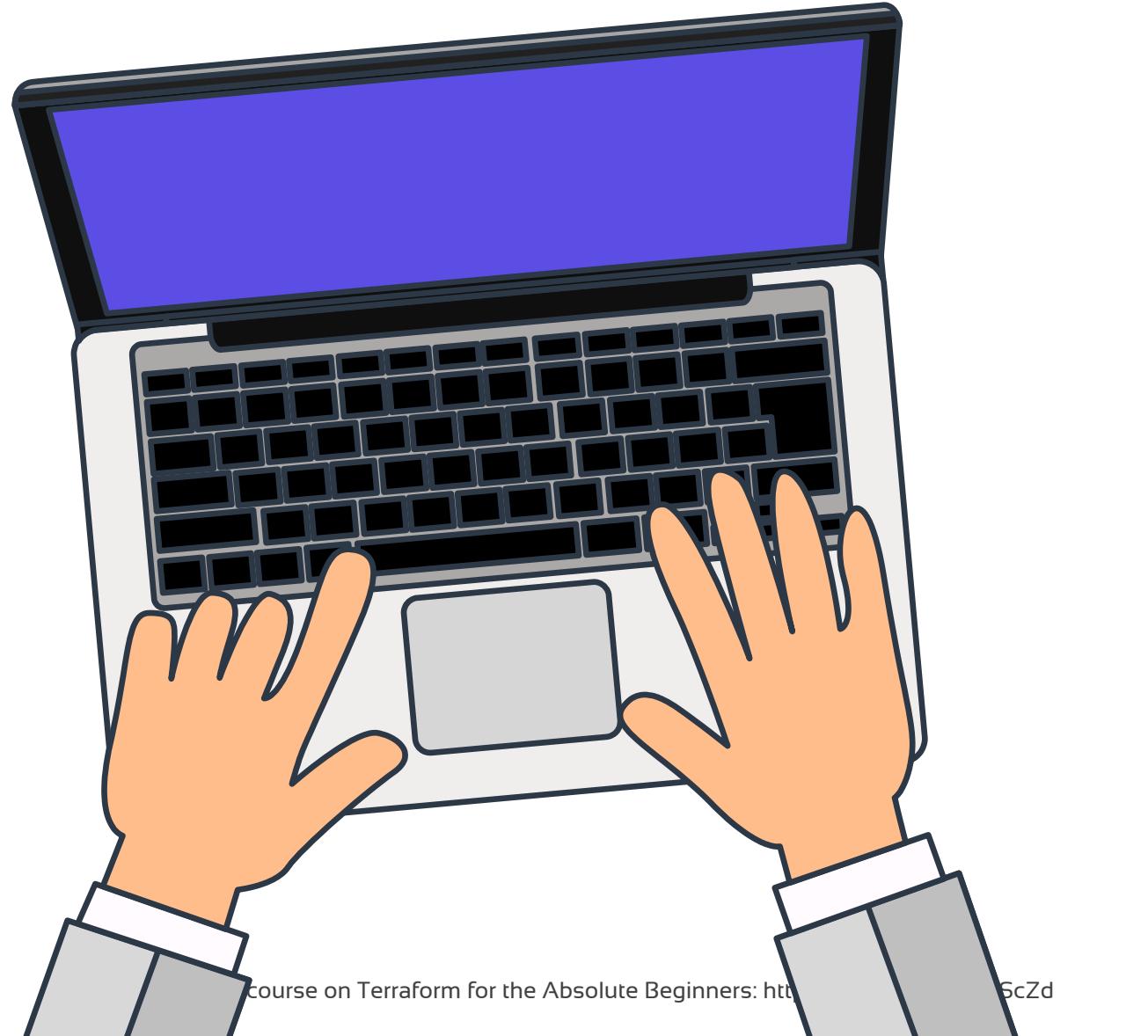
Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

local_file.pet: Destroying... [id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]
local_file.pet: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
```

HANDS-ON LABS



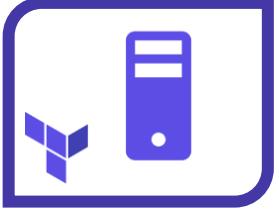
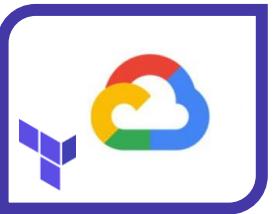


Terraform Basics

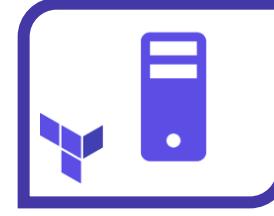
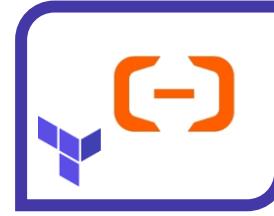
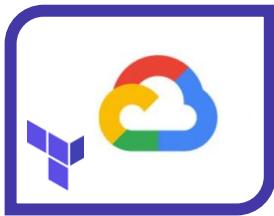
Using Terraform Providers

```
>_
```

```
$ terraform init
```



Official



Verified



bigip

by: F5Networks



heroku

by: heroku



[digitalocean](#)

by: [digitalocean](#)

Community



activedirectory



ucloud



netapp-gcp

```
>_
```

```
$ terraform init
```

```
Initializing the backend...
```

```
Initializing provider plugins...
```

```
- Finding latest version of hashicorp/local...
```

```
[ - Installing hashicorp/local v2.0.0...
```

```
[ - Installed hashicorp/local v2.0.0 (signed by HashiCorp)
```

```
The following providers do not have any version constraints in  
configuration,  
so the latest version was installed.
```

```
To prevent automatic upgrades to new major versions that may  
contain breaking  
changes, we recommend adding version constraints in a  
required_providers block  
in your configuration, with the constraint strings suggested  
below.
```

```
* hashicorp/local: version = "~> 2.0.0"
```

```
Terraform has been successfully initialized!
```

```
>_
```

```
$ ls /root/terraform-local-file/.terraform  
plugins
```

>_

```
$ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/local...
- Installing hashicorp/local v2.0.0...
- Installed hashicorp/local v2.0.0 (signed by HashiCorp)

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking changes, we recommend adding version constraints in a required_providers block in your configuration, with the constraint strings suggested below.

```
* hashicorp/local: version = "~> 2.0.0"
```

Terraform has been successfully initialized!

To prevent automatic upgrades to new major versions from containing breaking changes, we recommend adding version constraints to your required_providers block in your configuration, with the constraint below.

```
*[{"hashicorp/local": version = "~> 2.0.0"}]
```

Organizational
Namespace

Type

Terraform has been successfully initialized!

To prevent automatic upgrades to new major versions from containing breaking changes, we recommend adding version constraints to your required_providers block in your configuration, with the constraints shown below.

```
* provider "aws" {  
    version = "2.2.0"  
}
```

Hostname

Organizational Namespace

Type

Terraform has been successfully initialized!

Initializing provider plugins...

- Finding latest version of hashicorp/local...
 - **Installing hashicorp/local v2.0.0...**
 - **Installed hashicorp/local v2.0.0 (signed by HashiCorp)**
-

The following providers do not have any version configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions containing breaking changes, we recommend adding version constraint required_providers block

Configuration Directory

```
>_
```

```
[terraform-local-file]$ ls /root/terraform-local-file  
local.tf
```

local.tf

```
resource "local_file" "pet" {  
  filename = "/root/pets.txt"  
  content = "We love pets!"  
}
```

cat.tf

```
resource "local_file" "cat" {  
  filename = "/root/cat.txt"  
  content = "My favorite pet is Mr. Whiskers"  
}
```

local.tf

cat.tf

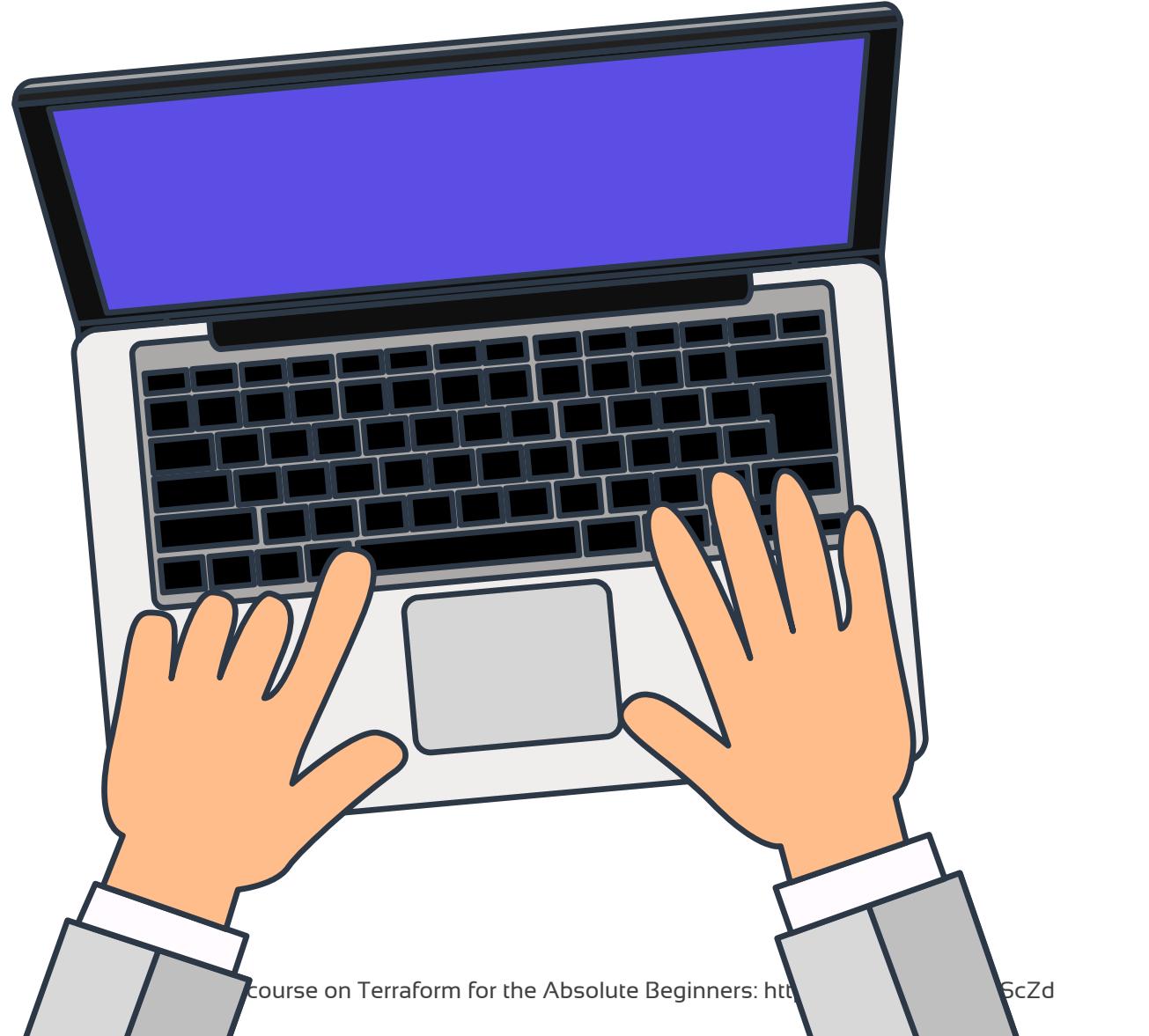
main.tf

```
resource "local_file" "pet" {
    filename = "/root/pets.txt"
    content = "We love pets!"
}

resource "local_file" "cat" {
    filename = "/root/cat.txt"
    content = "My favorite pet is Mr. Whiskers"
}
```

File Name	Purpose
main.tf	Main configuration file containing resource definition
variables.tf	Contains variable declarations
outputs.tf	Contains outputs from resources
provider.tf	Contains Provider definition

HANDS-ON LABS



Multiple Providers

main.tf

```
resource "local_file" "pet" {
  filename = "/root/pets.txt"
  content = "We love pets!"
}
```



```
main.tf

resource "local_file" "pet" {
    filename = "/root/pets.txt"
    content = "We love pets!"
}

resource "random_pet" "my-pet" {
    prefix = "Mrs"
    separator = "."
    length = "1"
}
```



random provider

Resources

random_id

random_integer

random_password

random_pet

random_shuffle

random_string

random_uuid



Argument Reference

The following arguments are supported:

- `keepers` - (Optional) Arbitrary map of values that, when provided, will be used to generate random values. See [the main provider documentation](#) for more information.
- `length` - (Optional) The length (in words) of the pet name.
- `prefix` - (Optional) A string to prefix the name with.
- `separator` - (Optional) The character to separate words in the generated name.

>_

```
$ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Using previously-installed hashicorp/local v2.0.0
- Finding latest version of hashicorp/random...
- Installing hashicorp/random v2.3.0...
- Installed hashicorp/random v2.3.0 (signed by HashiCorp)

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking changes, we recommend adding version constraints in a required_providers block in your configuration, with the constraint strings suggested below.

```
* hashicorp/local: version = "~> 2.0.0"
* hashicorp/random: version = "~> 2.3.0"
```

Terraform has been successfully initialized!



>_

```
$ terraform plan
```

```
Refreshing Terraform state in-memory prior to plan...
```

```
The refreshed state will be used to calculate this plan, but  
will not be  
persisted to local or remote state storage.
```

```
local_file.pet: Refreshing state...
```

```
[id=d1a31467f206d6ea8ab1cad382bc106bf46df69e]
```

```
.
```

```
.
```

```
# random_pet.my-pet will be created  
+ resource "random_pet" "my-pet" {  
    + id      = (known after apply)  
    + length   = 1  
    + prefix    = "Mrs"  
    + separator = "."  
}
```

```
Plan: 1 to add, 0 to change, 0 to destroy.
```



>_

```
$ terraform apply  
local_file.new_file: Refreshing state...  
[id=d1a31467f206d6ea8ab1cad382bc106bf46df69e]
```

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

```
# random_pet.my-pet will be created  
+ resource "random_pet" "my-pet" {  
    + id      = (known after apply)  
    + length   = 1  
    + prefix    = "Mrs"  
    + separator = "."  
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

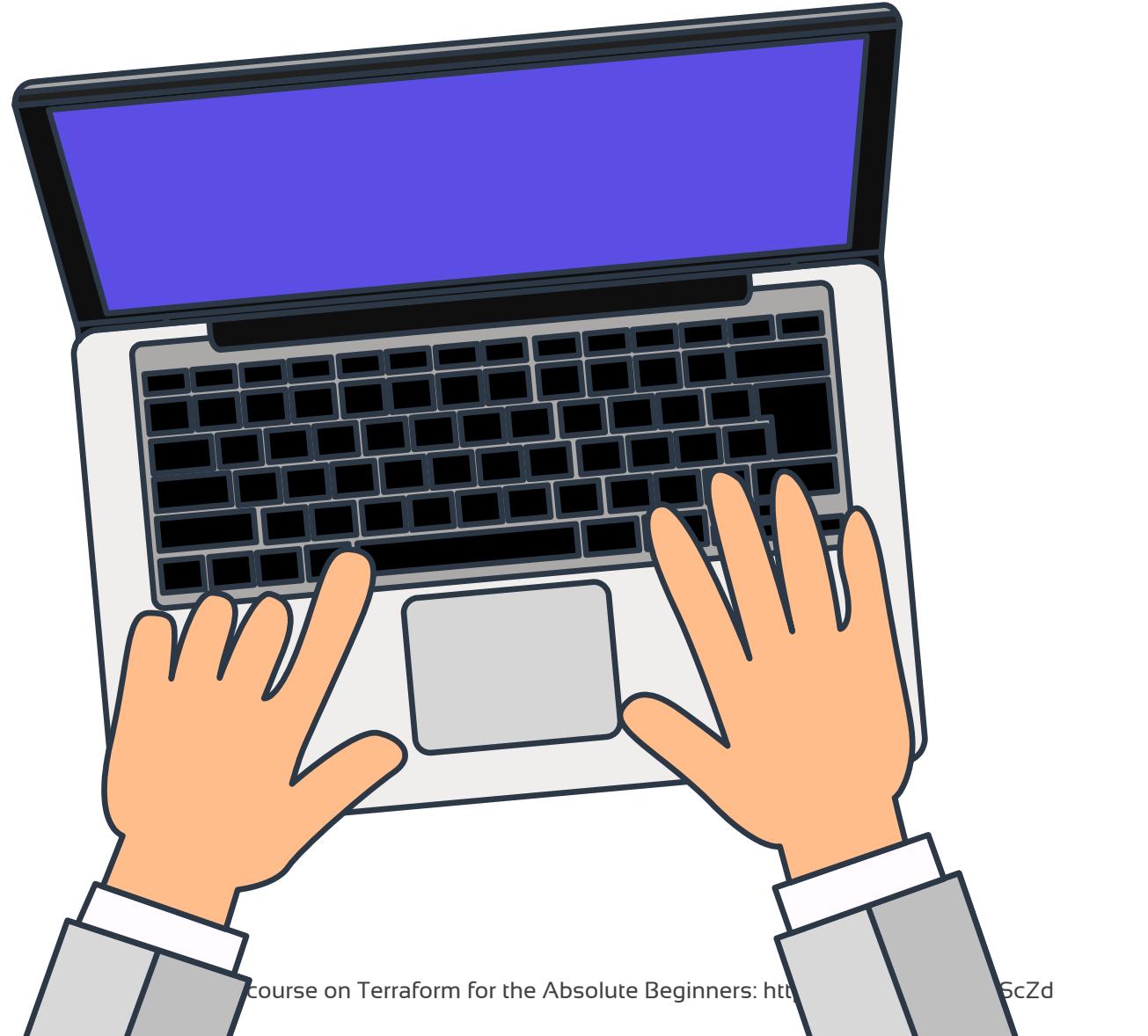
```
random_pet.my-pet: Creating...  
random_pet.my-pet: Creation complete after 0s [id=Mrs.hen]
```

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.



Mrs.hen

HANDS-ON LABS



Define Input Variables

main.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
}  
  
resource "random_pet" "my-pet" {  
    prefix = "Mrs"  
    separator = ":"  
    length = "1"  
}
```

Argument	Value
filename	"/root/pets.txt"
content	"We love pets!"
prefix	"Mrs"
separator	:
length	"1"

main.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
}  
  
resource "random_pet" "my-pet" {  
    prefix = "Mrs"  
    separator = "."  
    length = "1"  
}
```

variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
}  
variable "content" {  
    default = "We love pets!"  
}  
variable "prefix" {  
    default = "Mrs"  
}  
variable "separator" {  
    default = "."  
}  
variable "length" {  
    default = "1"  
}
```

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename  
    content = var.content  
}  
  
resource "random_pet" "my-pet" {  
    prefix = var.prefix  
    separator = var.separator  
    length = var.length  
}
```

variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
}  
variable "content" {  
    default = "We love pets!"  
}  
variable "prefix" {  
    default = "Mrs"  
}  
variable "separator" {  
    default = "."  
}  
variable "length" {  
    default = "1"  
}
```

>_

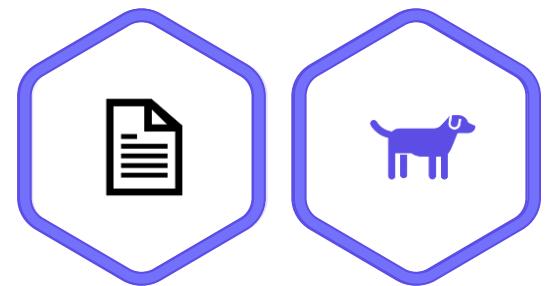
```
$ terraform apply

# local_file.pet will be created
+ resource "local_file" "pet" {
    + content          = "We love pets!"
    + directory_permission = "0777"
    + file_permission   = "0777"
    + filename         = "/root/pet.txt"
    + id               = (known after apply)
}

# random_pet.my-pet will be created
+ resource "random_pet" "my-pet" {
    + id           = (known after apply)
    + length       = 1
    + prefix       = "Mrs"
    + separator    = "."
}

Plan: 2 to add, 0 to change, 0 to destroy.

.
.
random_pet.my-pet: Creating...
random_pet.my-pet: Creation complete after 0s [id=Mrs.ram]
local_file.pet: Creating...
local_file.pet: Creation complete after 0s
[id=f392b4bcf5db76684f719bf72061627a9a177de1]
```



main.tf

```
resource "local_file" "pet" {  
    filename = var.filename  
    content = var.content  
}  
  
resource "random_pet" "my-pet" {  
    prefix = var.prefix  
    separator = var.separator  
    length = var.length  
}
```

variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
}  
variable "content" {  
    default = "My favorite pet is Mrs. Whiskers"  
}  
variable "prefix" {  
    default = "Mrs"  
}  
variable "separator" {  
    default = "."  
}  
variable "length" {  
    default = "2"  
}
```

```
>_
```

```
$ terraform apply
```

```
Terraform will perform the following actions:
```

```
-/+ resource "local_file" "pet" {
    ~ content          = "We love pets!" -> "My favorite pet is Mrs. Whiskers!" #
  forces replacement
    directory_permission = "0777"
    file_permission      = "0777"
    filename              = "/root/pet.txt"
    ~ id                 = "bc9cabef1d8b0071d3c4ae9959a9c328f35fe697" -> (known after
apply)
}

# random_pet.my-pet must be replaced
-/+ resource "random_pet" "my-pet" {
    ~ id           = "Mrs.Hen" -> (known after apply)
    ~ length       = 1 -> 2 # forces replacement
    prefix         = "Mrs"
    separator      = "."
}
```

```
Plan: 2 to add, 0 to change, 2 to destroy.
```

```
random_pet.my-pet: Destroying... [id=Mrs.hen]
```

```
random_pet.my-pet: Destruction complete after 0s
```

```
local_file.pet: Destroying... [id=bc9cabef1d8b0071d3c4ae9959a9c328f35fe697]
```

```
local_file.pet: Destruction complete after 0s
```

```
random_pet.my-pet: Creating...
```

```
local_file.pet: Creating...
```



main.tf

```
resource "aws_instance" "webserver" {  
    ami          = var.ami  
    instance_type = var.instance_type  
}
```

variables.tf

```
variable "ami" {  
    default = "ami-0edab43b6fa892279"  
}  
variable "instance_type" {  
    default = "t2.micro"  
}
```

Understanding the Variable Block

variables.tf

```
variable "filename" {
    default = "/root/pets.txt"
}
variable "content" {
    default = "I love pets!"
}
variable "prefix" {
    default = "Mrs"
}
variable "separator" {
    default = "."
}
variable "length" {
    default = "1"
}
```

variables.tf

```
variable "filename" {
    default = "/root/pets.txt"
    type = string
    description = "the path of local file"

}
variable "content" {
    default = "I love pets!"
    type = string
    description = "the content of the file"

}
variable "prefix" {
    default = "Mrs"
    type = string
    description = "the prefix to be set"

}
variable "separator" {
    default = "."
}
```

variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
    type = string  
    description = "the path of local file"  
}  
variable "content" {  
    default = "I love pets!"  
    type = string  
    description = "the content of the file"  
}  
variable "prefix" {  
    default = "Mrs"  
    type = string  
    description = "the prefix to be set"  
}  
variable "separator" {  
    default = "."
```

Type	Example
string	"/root/pets.txt"
number	1
bool	true/false
any	Default Value

variables.tf

```
variable "length" {  
    default = 2  
    type = number  
    description = "length of the pet name"  
}  
  
variable "password_change" {  
    default = true  
    type = bool  
}
```

Type	Example
string	"/root/pets.txt"
number	1
bool	true/false
any	Default Value
list	["cat", "dog"]
map	pet1 = cat pet2 = dog
object	Complex Data Structure
tuple	Complex Data Structure

List

variables.tf

```
variable "prefix" {  
  default = ["Mr", "Mrs", "Sir"]  
  type = list 0      1      2  
}
```

maint.tf

```
resource "random_pet" "my-pet" {  
  prefix      = var.prefix[0]  
}
```

Index	Value
0	Mr
1	Mrs
2	Sir

Map

variables.tf

```
variable file-content {  
    type      = map  
    default   = {  
        "statement1" = "We love pets!"  
        "statement2" = "We love animals!"  
    }  
}
```

maint.tf

```
resource local_file my-pet {  
    filename  = "/root/pets.txt"  
    content   = var.file-content["statement2"]  
}
```

Key	Value
statement1	We love pets!
statement2	We love animals!

List of a Type

variables.tf

```
variable "prefix" {  
    default = ["Mr", "Mrs", "Sir"]  
    type = list(string)  
}
```

variables.tf

```
variable "prefix" {  
    default = ["Mr", "Mrs", "Sir"]  
    type = list(number)  
}
```

variables.tf

```
variable "prefix" {  
    default = [1, 2, 3]  
    type = list(number)  
}
```

>_

```
$ terraform plan  
Error: Invalid default value for variable  
on variables.tf line 3, in variable "prefix":  
  3:   default      = ["Mr", "Mrs", "Sir"]  
  
This default value is not compatible with the  
variable's type constraint: a number is required.
```

Map of a Type

variables.tf

```
variable "cats" {  
  default = {  
    "color" = "brown"  
    "name" = "bella"  
  }  
  type = map(string)  
}
```

variables.tf

```
variable "pet_count" {  
  default = {  
    "dogs" = 3  
    "cats" = 1  
    "goldfish" = 2  
  }  
  type = map(number)  
}
```

Set

variables.tf

```
variable "prefix" {  
  default = ["Mr", "Mrs", "Sir"]  
  type = set(string)  
}
```

variables.tf

```
variable "prefix" {  
  default = ["Mr", "Mrs", "Sir", "Sir"]  
  type = set(string)  
}
```

variables.tf

```
variable "fruit" {  
  default = ["apple", "banana"]  
  type = set(string)  
}
```

variables.tf

```
variable "fruit" {  
  default = ["apple", "banana", "banana"]  
  type = set(string)  
}
```

variables.tf

```
variable "age" {  
  default = [10, 12, 15]  
  type = set(number)  
}
```

variables.tf

```
variable "age" {  
  default = [10, 12, 15, 10]  
  type = set(number)  
}
```

Objects

Key	Example	Type
name	bella	string
color	brown	string
age	7	number
food	["fish", "chicken", "turkey"]	list
favorite_pet	true	bool

```
variables.tf

variable "bella" {
  type = object({
    name = string
    color = string
    age = number
    food = list(string)
    favorite_pet = bool
  })

  default = {
    name = "bella"
    color = "brown"
    age = 7
    food = ["fish", "chicken", "turkey"]
    favorite_pet = true
  }
}
```

Tuples

variables.tf

```
variable kitty {  
  type      = tuple([string, number, bool])  
  default   = ["cat", 7, true]  
}
```

variables.tf

```
variable kitty {  
  type      = tuple([string, number, bool])  
  default   = ["cat", 7, true, "dog"]  
}
```

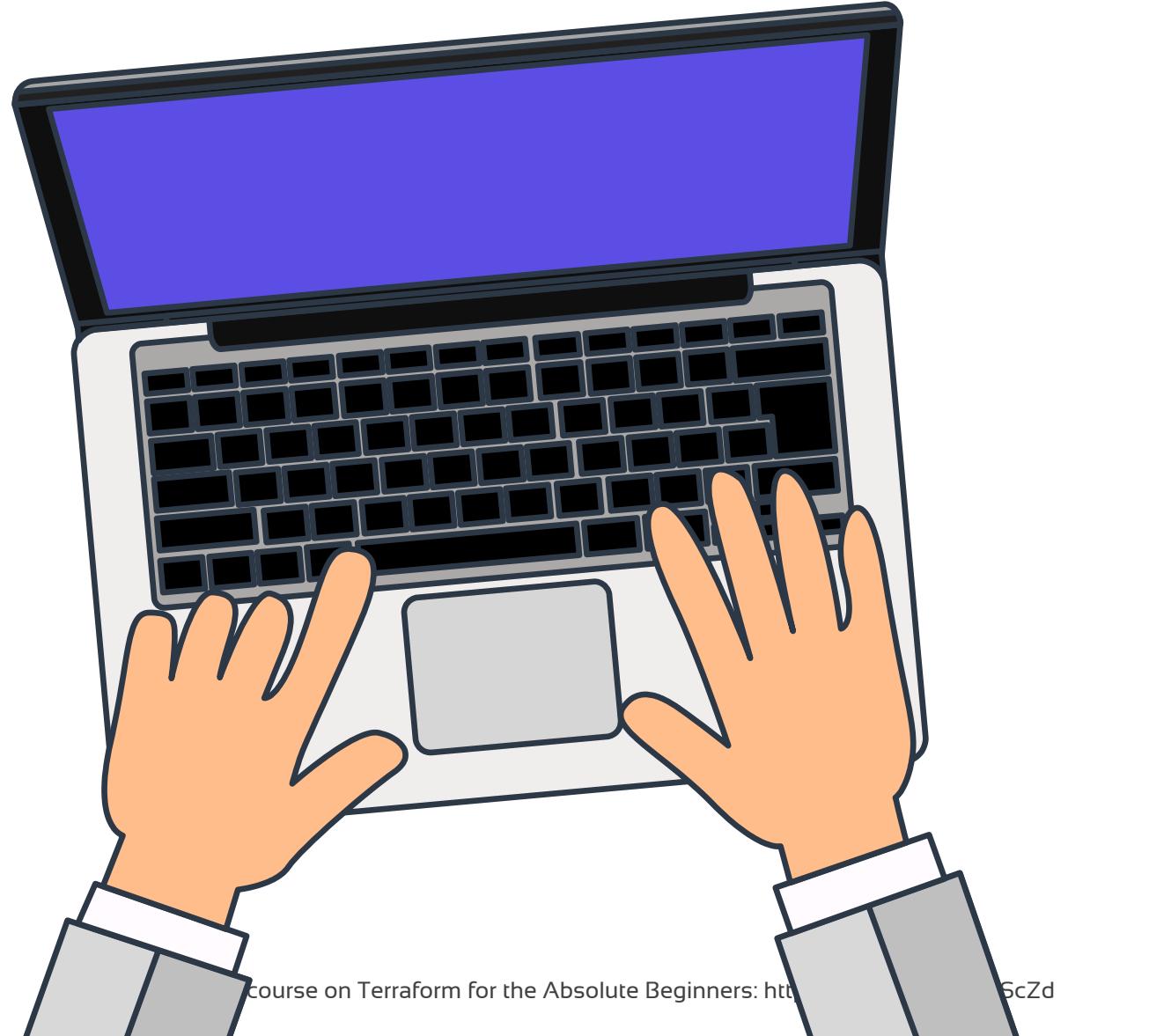
>_

\$ terraform plan

```
Error: Invalid default value for variable  
on variables.tf line 3, in variable "kitty":  
  3:   default      = ["cat", 7, true, "dog"]
```

This default value is not compatible with the
variable's type constraint:
tuple required.

HANDS-ON LABS



Using Variables in Terraform

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename  
    content = var.content  
}  
  
resource "random_pet" "my-pet" {  
    prefix = var.prefix  
    separator = var.separator  
    length = var.length  
}
```

variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
}  
variable "content" {  
    default = "We love pets!"  
}  
variable "prefix" {  
    default = "Mrs"  
}  
variable "separator" {  
    default = "."  
}  
variable "length" {  
    default = 2  
}
```

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename  
    content = var.content  
}  
  
resource "random_pet" "my-pet" {  
    prefix = var.prefix  
    separator = var.separator  
    length = var.length  
}
```

variables.tf

```
variable "filename" {  
}  
variable "content" {  
}  
variable "prefix" {  
}  
variable "separator" {  
}  
variable "length" {  
}
```

Interactive Mode

```
>_
$ terraform apply
var.content
  Enter a value: We love Pets!

var.filename
  Enter a value: /root/pets.txt

var.length
  Enter a value: 2

var.prefix
  Enter a value: Mrs.

var.separator
  Enter a value: .
```

Command Line Flags

```
>_
```

```
$ terraform apply -var "filename=/root/pets.txt" -var "content=We love  
Pets!" -var "prefix=Mrs" -var "separator=." -var "length=2"
```

Environment Variables

```
>_  
  
$ export TF_VAR_filename="/root/pets.txt"  
$ export TF_VAR_content="We love pets!"  
$ export TF_VAR_prefix="Mrs"  
$ export TF_VAR_separator=". "  
$ export TF_VAR_length="2"  
$ terraform apply
```

Variable Definition Files

```
terraform.tfvars
```

```
filename = "/root/pets.txt"
content = "We love pets!"
prefix = "Mrs"
separator = "."
length = "2"
```

```
>_
```

```
$ terraform apply -var-file variables.tfvars
```

terraform.tfvars

terraform.tfvars.json

*.auto.tfvars

*.auto.tfvars.json

Automatically Loaded

Variable Definition Precedence

main.tf

```
resource local_file pet {  
    filename = var.filename  
}
```

variables.tf

```
variable filename {  
    type    = string  
}
```

>_

```
$ export TF_VAR_filename="/root/cats.txt" ?
```

terraform.tfvars

```
filename = "/root/pets.txt" ?
```

>_

variable.auto.tfvars

```
filename = "/root/mypet.txt" ?
```

>_

```
$ terraform apply -var "filename=/root/best-pet.txt" ?
```

Variable Definition Precedence

Order	Option
1	Environment Variables
2	terraform.tfvars
3	*.auto.tfvars (alphabetical order)
4	-var or --var-file (command-line flags)



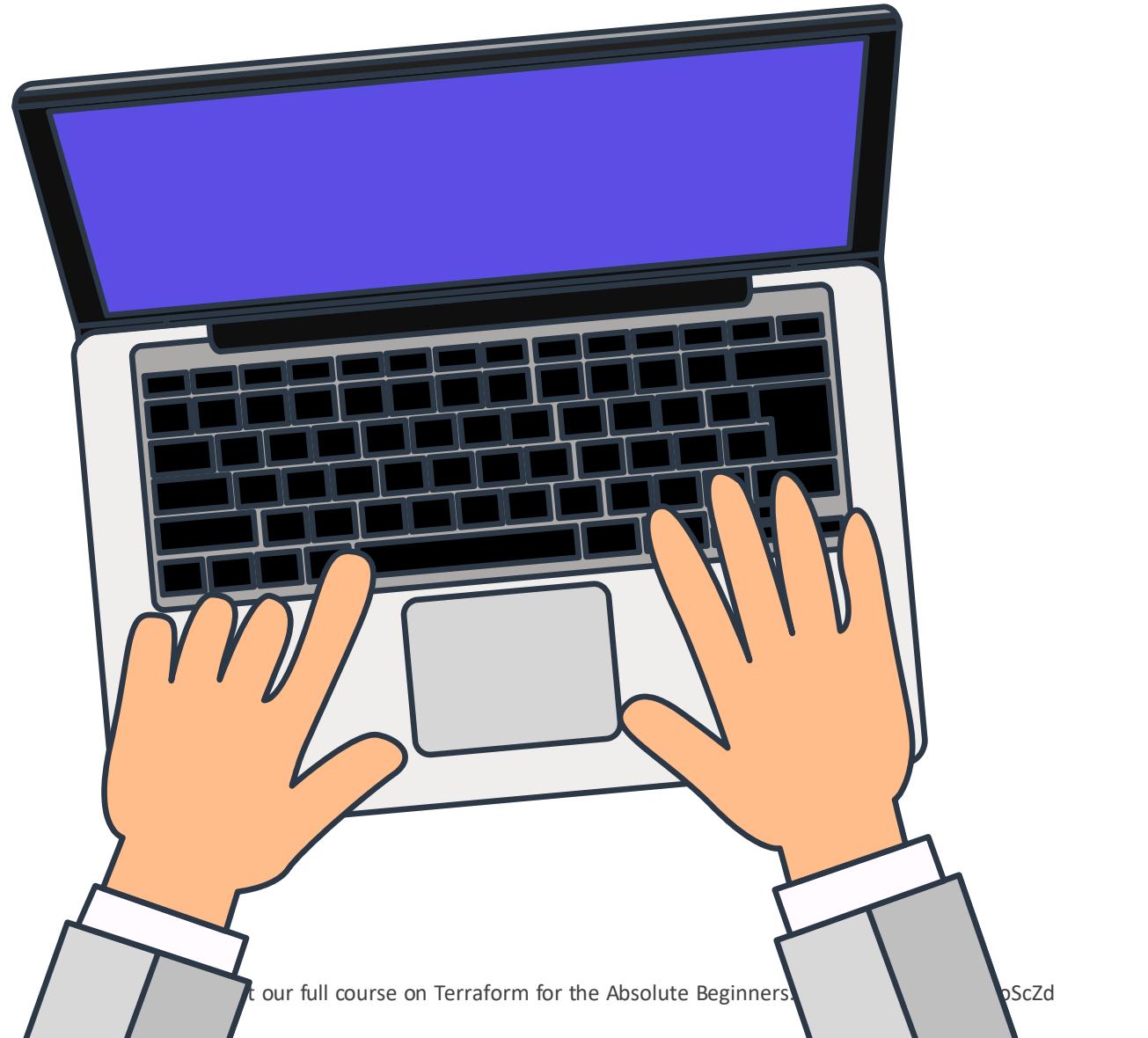
```
>_
$ export TF_VAR_filename="/root/cats.txt" 1
```

```
●      terraform.tfvars
filename = "/root/pets.txt" 2
```

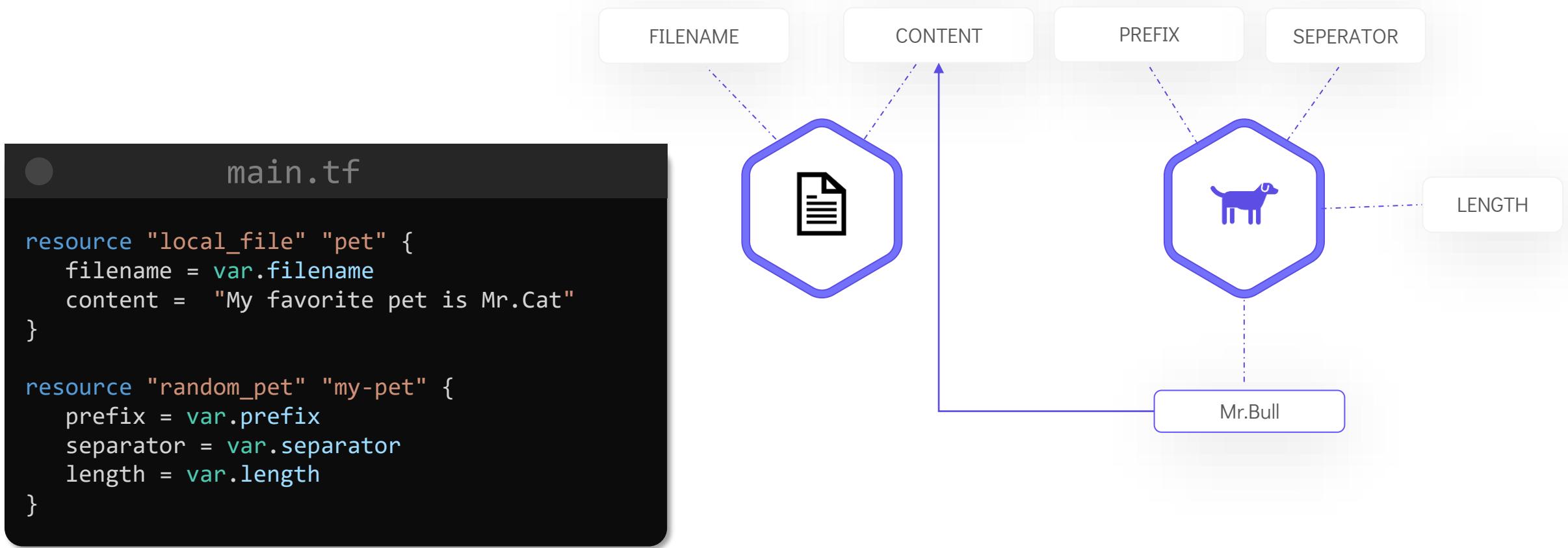
```
●      variable.auto.tfvars
filename = "/root/mypet.txt" 3
```

```
>_
$ terraform apply -var "filename=/root/best-pet.txt" 4
```

HANDS-ON LABS



Resource Attribute Reference



```

>_
random_pet.my-pet: Creating...
local_file.pet: Creating...
random_pet.my-pet: Creation complete after 0s [id=Mr.bull]
local_file.pet: Creation complete after 0s [id=059090e865809f9b6debfd7aebf48fdce2220a6]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

```

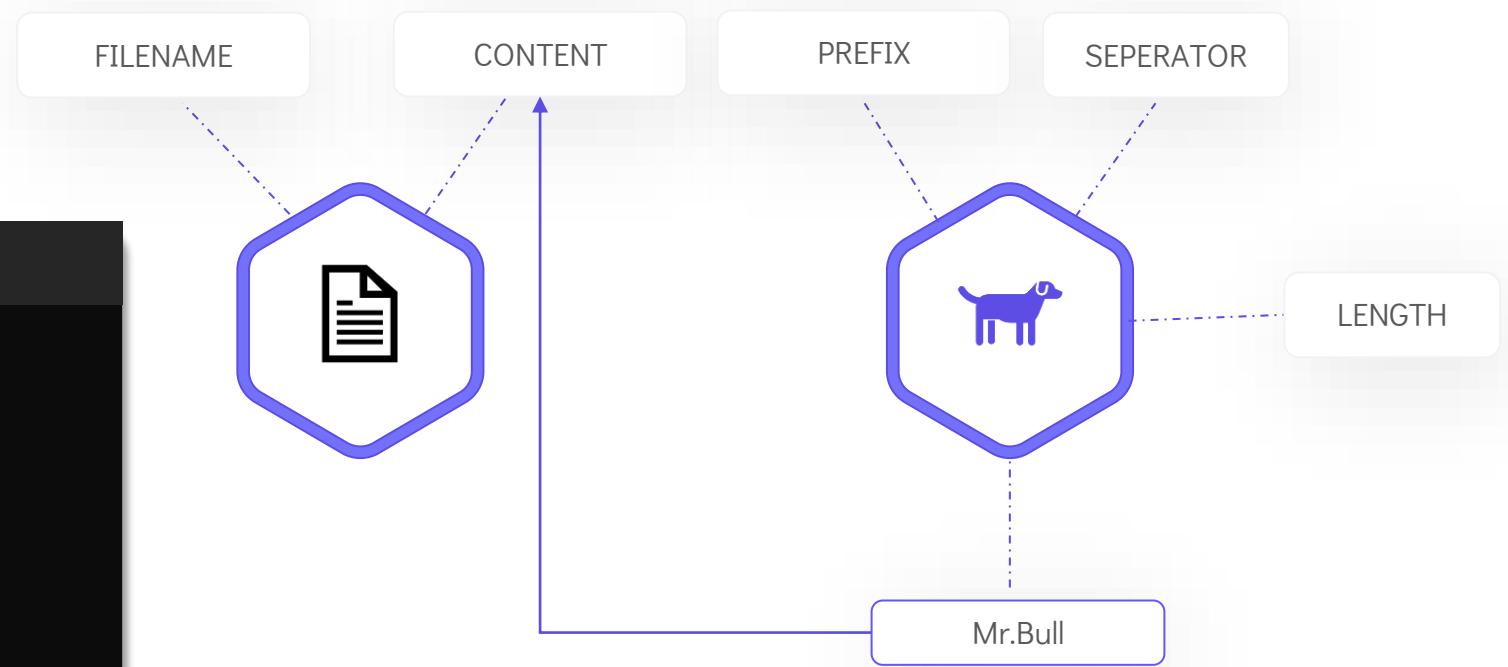
```

main.tf

resource "local_file" "pet" {
  filename = var.filename
  content = "My favorite pet is Mr.Cat"
}

resource "random_pet" "my-pet" {
  prefix = var.prefix
  separator = var.separator
  length = var.length
}

```



```

>_
random_pet.my-pet: Creating...
local_file.pet: Creating...
random_pet.my-pet: Creation complete after 0s [id=Mr.bull]
local_file.pet: Creation complete after 0s [id=059090e865809f9b6debfd7aebf48fdce2220a6]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

```

Attribute Reference

The following attributes are supported:

- `id` - (string) The random pet name

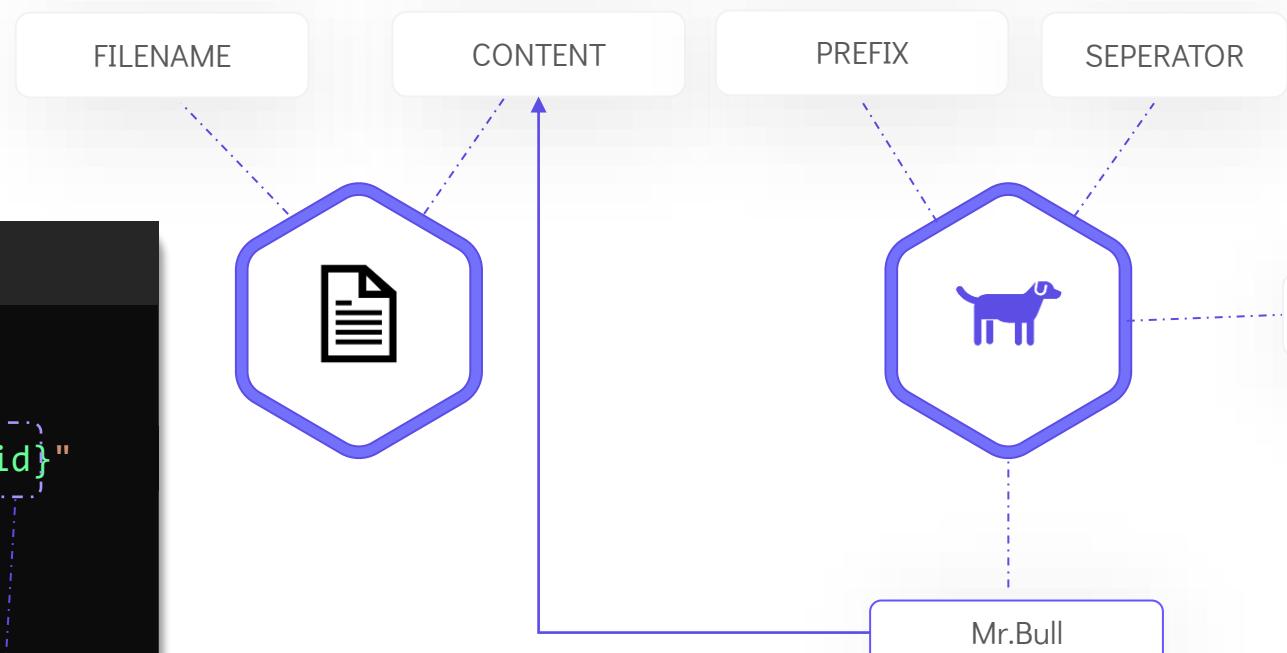
```

main.tf

resource "local_file" "pet" {
  filename = var.filename
  content = "My favorite pet is ${random_pet.my-pet.id}"
}

resource "random_pet" "my-pet" {
  prefix = var.prefix
  separator = var.separator
  length = var.length
}

```



```

>_
random_pet.my-pet: Creating...
local_file.pet: Creating...
random_pet.my-pet: Creation complete after 0s [id=Mr.bull]
local_file.pet: Creation complete after 0s
[id=059090e865809f9b6debfd7aebf48fdce2220a6]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

```

Attribute Reference

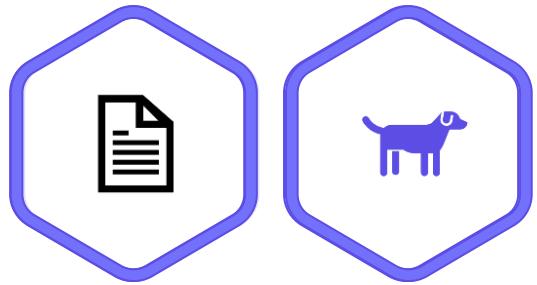
The following attributes are supported:

- `id` - (string) The random pet name

```
_file" "pet" {  
var.filename
```

My favorite pet is Mr.Bull"

```
om_pet" "my-pet" {  
.prefix  
var.separator  
.length
```



```
>_ $ terraform apply

.

.

.

# local_file.pet must be replaced
-/+ resource "local_file" "pet" {
  ~ content          = "My favorite pet is Mrs.Cat!" ->
  "My favorite pet is Mr.bull" # forces replacement
  ~ directory_permission = "0777"
  file_permission       = "0777"
  filename              = "/roots/pets.txt"
  ~ id                  =
  "98af5244e23508cffd4a0c3c46546821c4ccb0" -> (known after
apply)
}

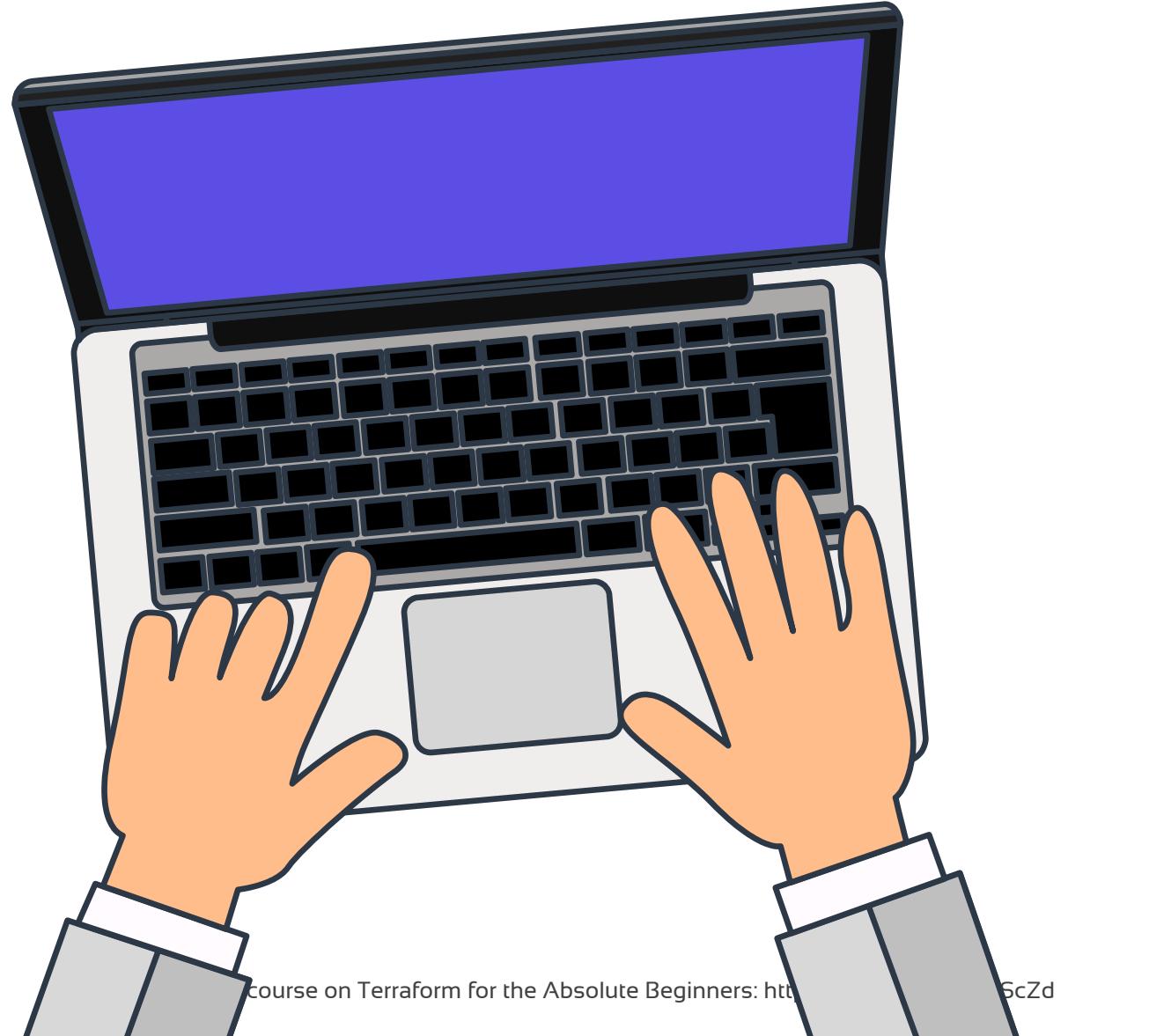
.

.

local_file.pet: Destroying...
[id=98af5244e23508cffd4a0c3c46546821c4ccb0]
local_file.pet: Destruction complete after 0s
local_file.pet: Creating...
local_file.pet: Creation complete after 0s
[id=e56101d304de7cf1b1001102923c6bdeaa60c523]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
```

HANDS-ON LABS



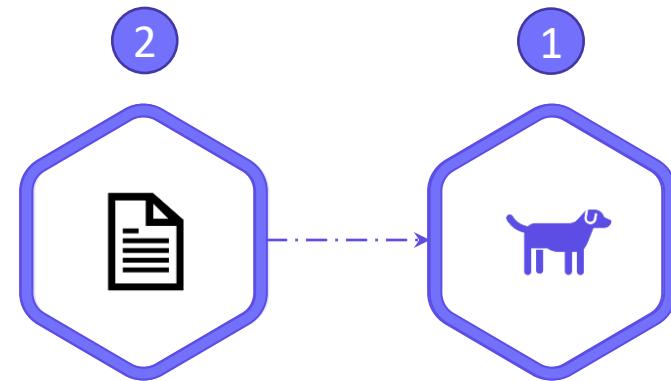
Resource Dependencies

Implicit Dependency

main.tf

```
resource "local_file" "pet" {
  filename = var.filename
  content = "My favorite pet is ${random_pet.my-pet.id}"
}

resource "random_pet" "my-pet" {
  prefix = var.prefix
  separator = var.separator
  length = var.length
}
```

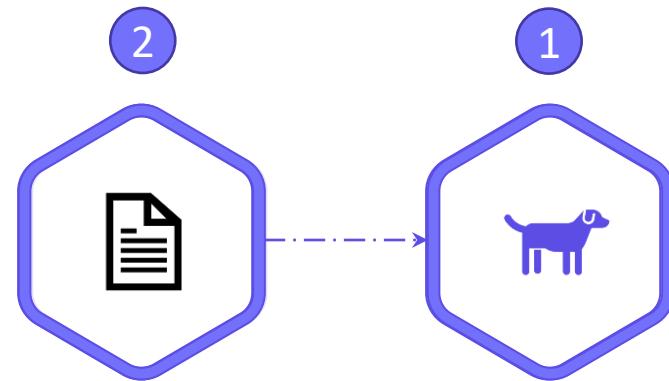


Explicit Dependency

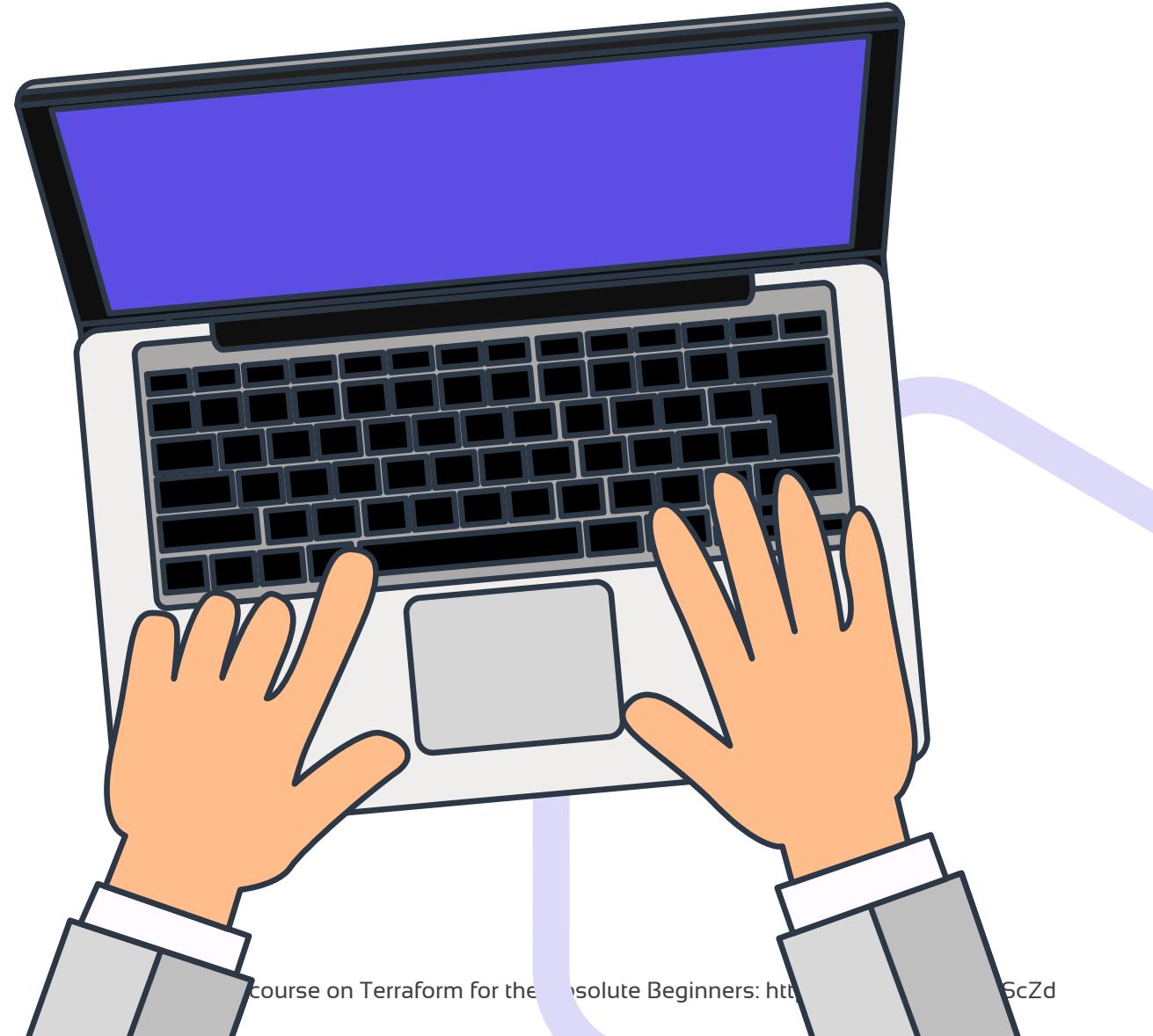
```
main.tf

resource "local_file" "pet" {
  filename = var.filename
  content = "My favorite pet is Mr.Cat"
  depends_on = [
    random_pet.my-pet
  ]
}

resource "random_pet" "my-pet" {
  prefix = var.prefix
  separator = var.separator
  length = var.length
}
```



HANDS-ON LABS



Output Variables

main.tf

```
resource "local_file" "pet" {
  filename = var.filename
  content = "My favorite pet is ${random_pet.my-pet.id}"
}

resource "random_pet" "my-pet" {
  prefix = var.prefix
  separator = var.separator
  length = var.length
}

output pet-name {
  value      = random_pet.my-pet.id
  description = "Record the value of pet ID generated by the
random_pet resource"
}
```

variables.tf

```
variable "filename" {
  default = "/root/pets.txt"
}
variable "content" {
  default = "I love pets!"
}
variable "prefix" {
  default = "Mrs"
}
variable "separator" {
  default = "."
}
variable "length" {
  default = "1"
}
```

```
output "<variable_name>" {
  value = "<variable_value>"
  <arguments>
}
```



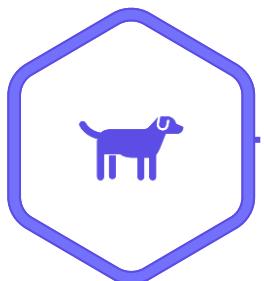
```
>_
$ terraform apply
:
:
Outputs:
pet-name = Mrs.gibbon
```

```
>_
```

```
$ terraform output  
pet-name = Mrs.gibbon
```

```
>_
```

```
$ terraform output pet-name  
Mrs.gibbon
```



Output Variable

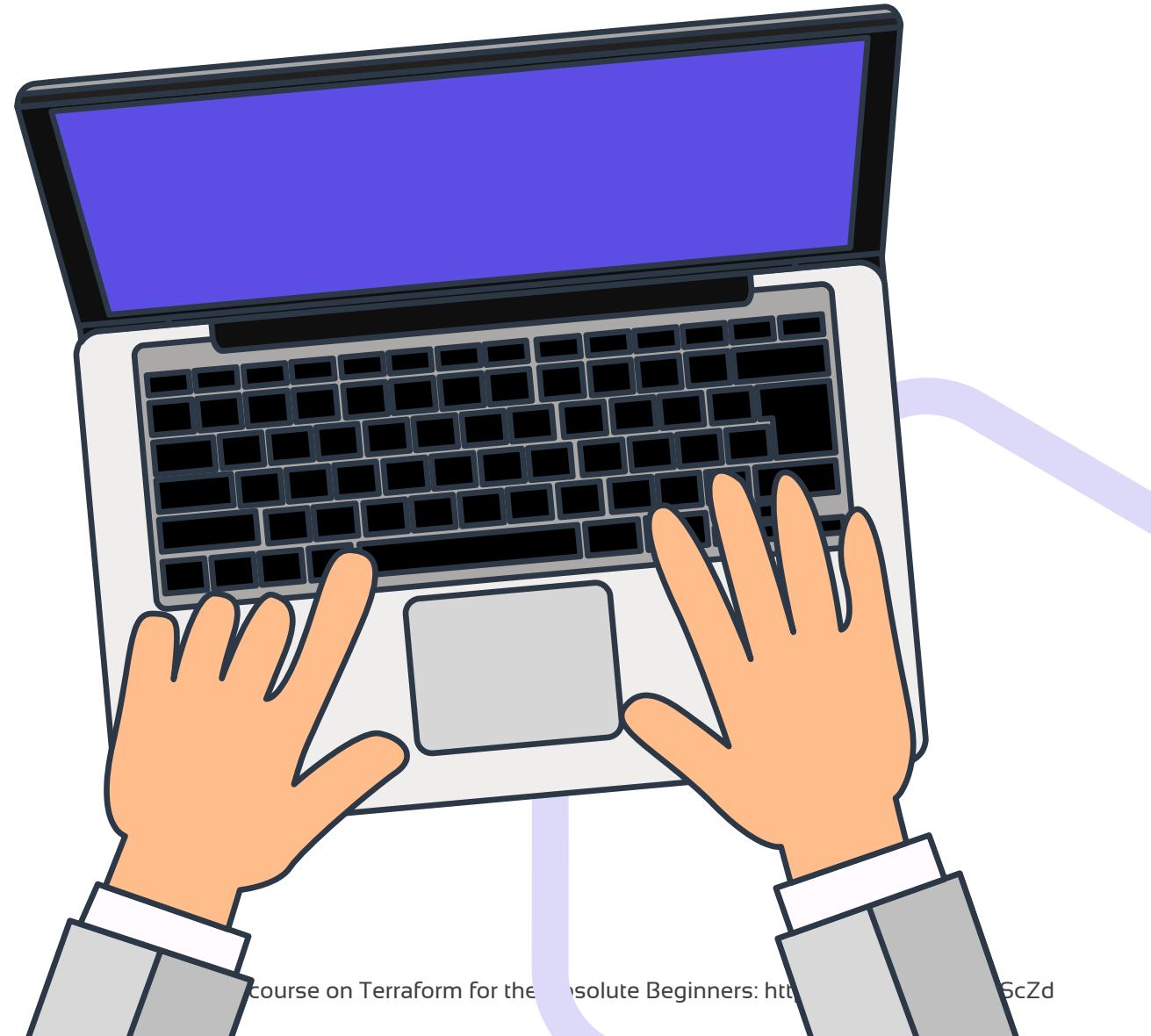


ANSIBLE



SHELL SCRIPTS

HANDS-ON LABS



Introduction to Terraform State

```
>_
```

```
$ ls terraform-local-file  
main.tf variables.tf
```



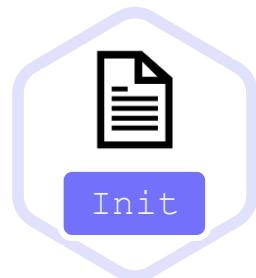
main.tf

```
resource "local_file" "pet" {  
    filename = var.filename  
    content  = var.content  
}
```



variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
}  
variable "content" {  
    default = "I love pets!"  
}
```



```
>_
```

```
$ cd terraform-local-file  
[terraform-local-file]$ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/local...
- Installing hashicorp/local v1.4.0...
- Installed hashicorp/local v1.4.0 (signed by HashiCorp)

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking changes, we recommend adding version constraints in a required_providers block in your configuration, with the constraint strings suggested below.

* hashicorp/local: version = "~> 1.4.0"

Terraform has been successfully initialized!

```
>_
```

```
$ ls terraform-local-file
```

```
main.tf variables.tf
```

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename  
    content  = var.content  
}
```

variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
}  
variable "content" {  
    default = "I love pets!"  
}
```



```
>_
```

```
[terraform-local-file]$ terraform plan
```

Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan,
persisted to local or remote state storage.

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
`+ create`

Terraform will perform the following actions:

```
# local_file.pet will be created  
+ resource "local_file" "pet" {  
    + content          = "I love pets!"  
    + directory_permission = "0777"  
    + file_permission   = "0777"  
    + filename         = "/root/pets.txt"  
    + id               = (known after apply)  
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

```
>_
```

```
$ ls terraform-local-file  
main.tf variables.tf
```

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename  
    content  = var.content  
}
```

variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
}  
variable "content" {  
    default = "I love pets!"  
}
```



```
>_
```

```
[terraform-local-file]$ terraform apply
```

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

```
# local_file.pet will be created  
+ resource "local_file" "pet" {  
    + content          = "I love pets!"  
    + directory_permission = "0777"  
    + file_permission   = "0777"  
    + filename         = "/root/pets.txt"  
    + id               = (known after apply)  
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

```
local_file.pet: Creating...
```

```
local_file.pet: Creation complete after 0s
```

```
[id=7e4db4fbfdbb108bdd04692602bae3e9bd1e1b68] !
```

>_

```
[terraform-local-file]$ cat /root/pets  
I love pets!
```

>_

```
[terraform-local-file]$ terraform apply  
local_file.pet: Refreshing state...  
[id=7e4db4fbfdbb108bdd04692602bae3e9bd1e1b68]
```

```
Apply complete! Resources: 0 added, 0 changed, 0 destroyed
```



```
>_
```

```
[terraform-local-file]$ ls  
main.tf variables.tf terraform.tfstate
```



```
>_
```

```
[terraform-local-file]$ cat terraform.tfstate
```

```
{  
    "version": 4,  
    "terraform_version": "0.13.0",  
    "serial": 1,  
    "lineage": "e35dde72-a943-de50-3c8b-1df8986e5a31",  
    "outputs": {},  
    "resources": [  
        {  
            "mode": "managed",  
            "type": "local_file",  
            "name": "pet",  
            "provider":  
                "provider[\"registry.terraform.io/hashicorp/local\"]",  
            "instances": [  
                {  
                    "schema_version": 0,  
                    "attributes": {  
                        "content": "I love pets!",  
                        "content_base64": null,  
                        "directory_permission": "0777",  
                        "file_permission": "0777",  
                        "filename": "/root/pets.txt",  
                        "id":  
                            "7e4db4fbfdbb108bdd04692602bae3e9bd1e1b68",  
                        "sensitive_content": null  
                    },  
                    "private": "bnVsbA=="  
                }  
            ]  
        }  
    ]  
}
```

variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
}  
variable "content" {  
    default = "We love pets!"  
}
```

```
>_ $ terraform plan  
$
```

Refreshing Terraform state in-memory
prior to plan...
The refreshed state will be used to
calculate this plan, but will not be
persisted to local or remote state
storage.

```
local_file.pet: Refreshing state...  
[id=7e4db4fbfdbb108bdd04692602bae3e9bd1e  
1b68]  
.  
.  
.  
.
```

[Output Truncated]



>_

```
[terraform-local-file]$ cat terraform.tfstate
```

```
{  
    "version": 4,  
    "terraform_version": "0.13.0",  
    "serial": 1,  
    "lineage": "e35dde72-a943-de50-3c8b-1df8986e5a31",  
    "outputs": {},  
    "resources": [  
        {  
            "mode": "managed",  
            "type": "local_file",  
            "name": "pet",  
            "provider":  
                "provider": "registry.terraform.io/hashicorp/local",  
                "instances": [  
                    {  
                        "schema_version": 0,  
                        "attributes": {  
                            "content": "I love pets!",  
                            "content_base64": null,  
                            "directory_permission": "0777",  
                            "file_permission": "0777",  
                            "filename": "/root/pets.txt",  
                            "id": "7e4db4fbfdbb108bdd04692602bae3e9bd1e1b68",  
                            "sensitive_content": null  
                        },  
                        "private": "bnVsbA=="  
                    }  
                ]  
        }  
    ]  
}
```

Check

variables.tf

```
variable "filename" {
  default = "/root/pets.txt"
}
variable "content" {
  default = "We love pets!"
}
```

>_

```
$ terraform apply
```

```
local_file.pet: Refreshing state...
[ id=7e4db4fbfdbb108bdd04692602bae3e9bd1e1b68 ]
```

```
Terraform will perform the following actions:
```

```
# local_file.pet must be replaced
-/+ resource "local_file" "pet" {
    ~ content          = "I love pets!" -
> "We love pets!" # forces replacement
    directory_permission = "0777"
    file_permission      = "0777"
    filename             = "/root/pets.txt"
    ~ id                 =
"7e4db4fbfdbb108bdd04692602bae3e9bd1e1b68" ->
(known after apply)
}
```



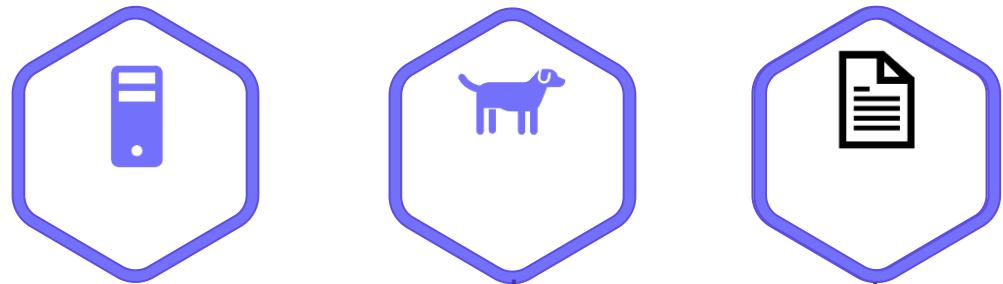
>_

```
[terraform-local-file]$ cat terraform.tfstate
```

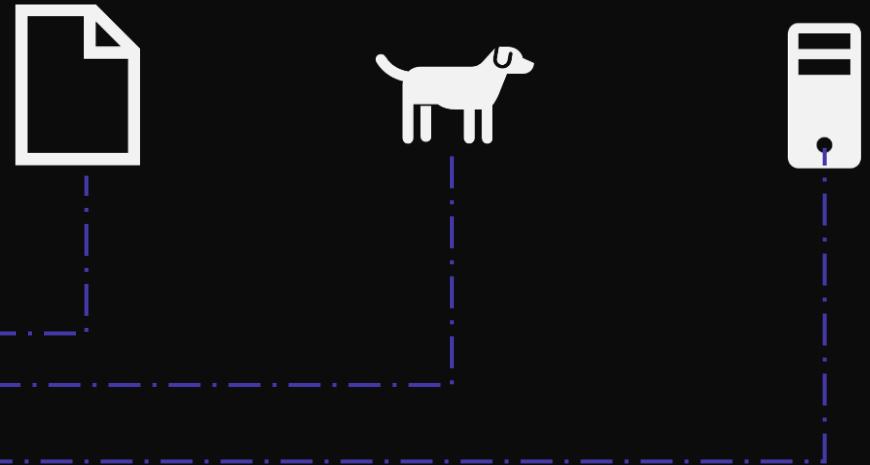
```
{
  "version": 4,
  "terraform_version": "0.13.0",
  "serial": 1,
  "lineage": "e35dde72-a943-de50-3c8b-1df8986e5a31",
  "outputs": {},
  "resources": [
    {
      "mode": "managed",
      "type": "local_file",
      "name": "pet",
      "provider": "provider[\\"registry.terraform.io/hashicorp/local\\"]",
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "content": "We love pets!",
            "content_base64": null,
            "directory_permission": "0777",
            "file_permission": "0777",
            "filename": "/root/pets.txt",
            "id": "7e4db4fbfdbb108bdd04692602bae3e9bc4d1c14",
            "sensitive_content": null
          },
          "private": "bnVsbA=="
        }
      ]
    }
  ]
}
```

Check

Real World Infrastructure

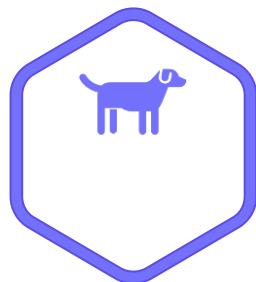
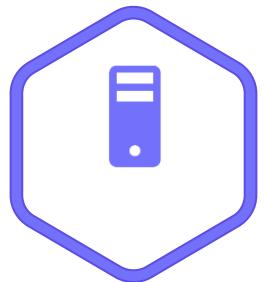


terraform.tfstate



Purpose of State

Real World Infrastructure



terraform.tfstate



id=aabbcc

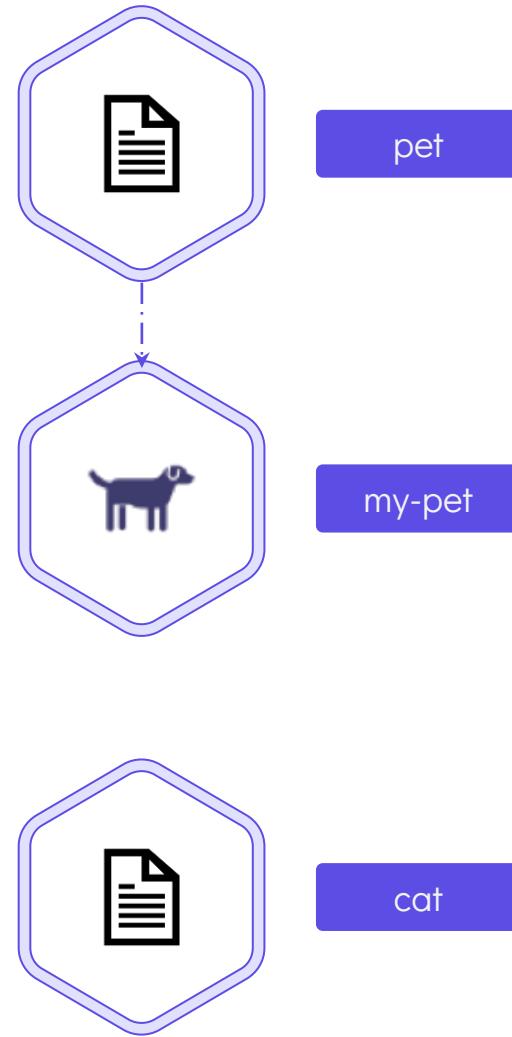
id=eeddff

id=gghhhii

Tracking Metadata

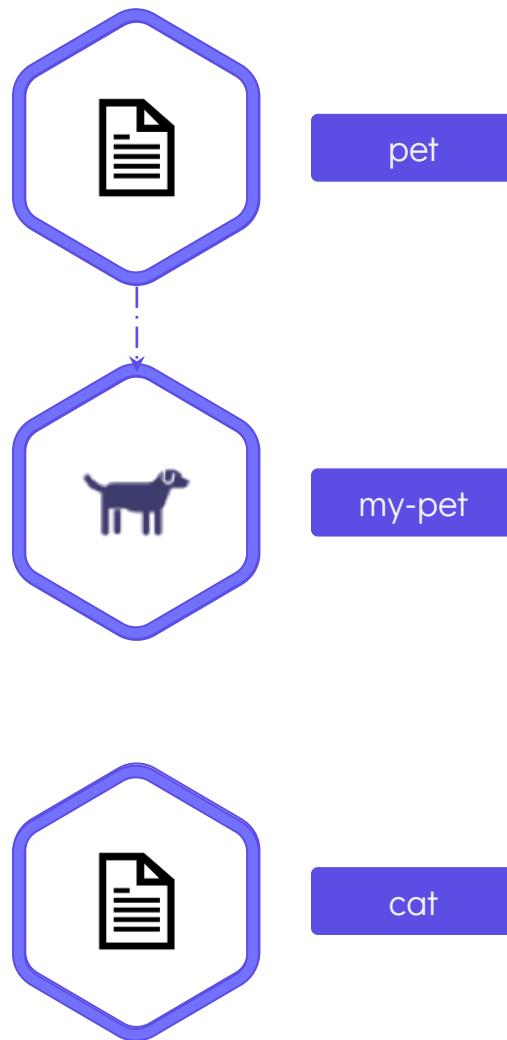
```
main.tf
```

```
resource "local_file" "pet" {
  filename = "/root/pet.txt"
  content  = "My favorite pet is ${random_pet.my-pet.id}!"
}
resource "random_pet" "my-pet" {
  length = 1
}
resource "local_file" "cat" {
  filename = "/root/cat.txt"
  content  = "I like cats too!"
}
```



Tracking Metadata

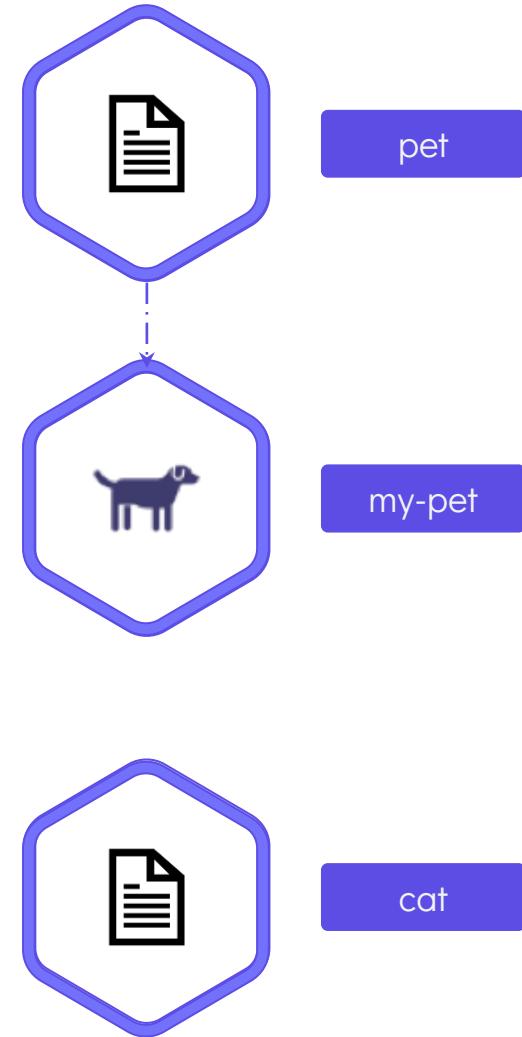
```
>_  
$ terraform apply  
. . .  
Plan: 3 to add, 0 to change, 0 to destroy.  
  
Do you want to perform these actions?  
Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.  
  
Enter a value: yes  
local_file.cat: Creating...  
random_pet.my-pet: Creating...  
local_file.cat: Creation complete after 0s  
[id=fe44888891fc40342313bc44a1f1a8986520c89]  
random_pet.my-pet: Creation complete after 0s [id=yak]  
  
local_file.pet: Creating...  
local_file.pet: Creation complete after 0s  
[id=28b373c6c1fa3fce132a518eadd0175c98f37f20]  
  
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```



Tracking Metadata

```
main.tf
```

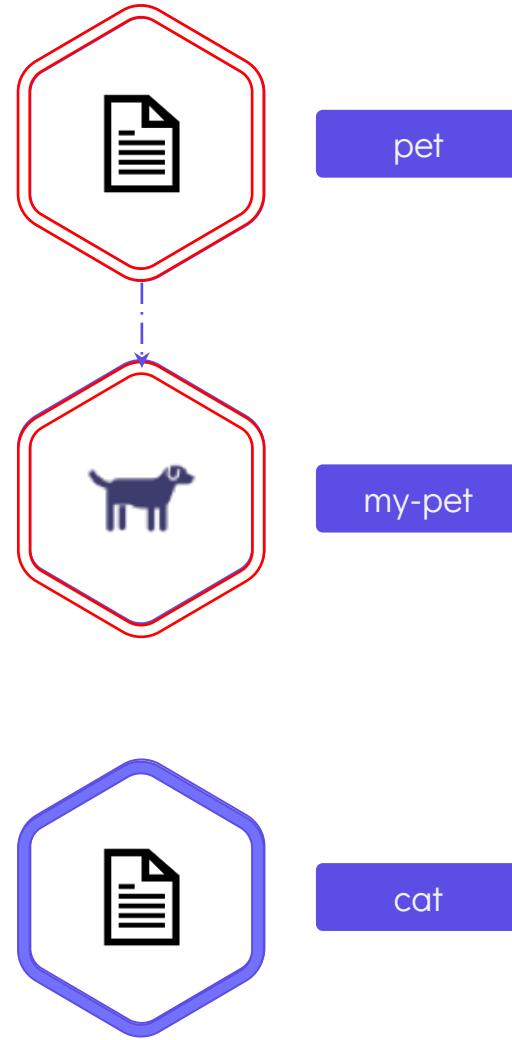
```
resource "local_file" "pet" {
  filename = "/root/pet.txt"
  content  = "My favorite pet is ${random_pet.my-pet.id}!"
}
resource "random_pet" "my-pet" {
  length = 1
}
resource "local_file" "cat" {
  filename = "/root/cat.txt"
  content  = "I like cats too!"
}
```



Tracking Metadata

main.tf

```
resource "local_file" "cat" {
  filename = "/root/cat.txt"
  content  = "I like cats too!"
}
```



Tracking Metadata

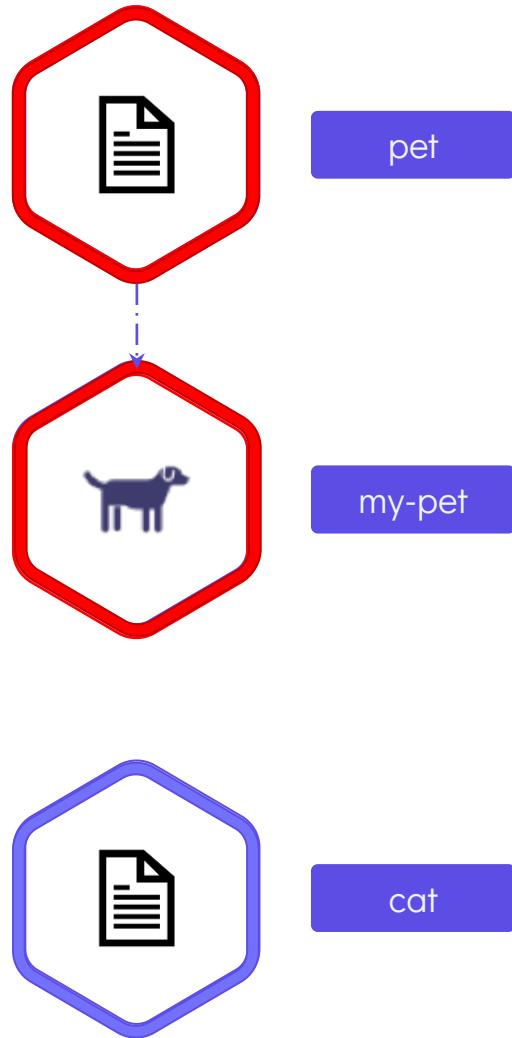
```
main.tf
```

```
resource "local_file" "cat" {
  filename = "/root/cat.txt"
  content  = "I like cats too!"
}
```

```
>_
```

```
$ cat terraform.tfstate
```

```
{
  "mode": "managed",
  "type": "local_file",
  "name": "pet",
  "instances": [
    {
      "schema_version": 0,
      "attributes": {
        "content": "My favorite pet is yak!",
      },
      "private": "bnVsbA==",
      "dependencies": [
        "random_pet.my-pet"
      ]
    }
  ]
}
```



Tracking Metadata

```
main.tf
```

```
resource "local_file" "cat" {
  filename = "/root/cat.txt"
  content  = "I like cats too!"
}
```

```
>_
```

```
$ terraform apply
```

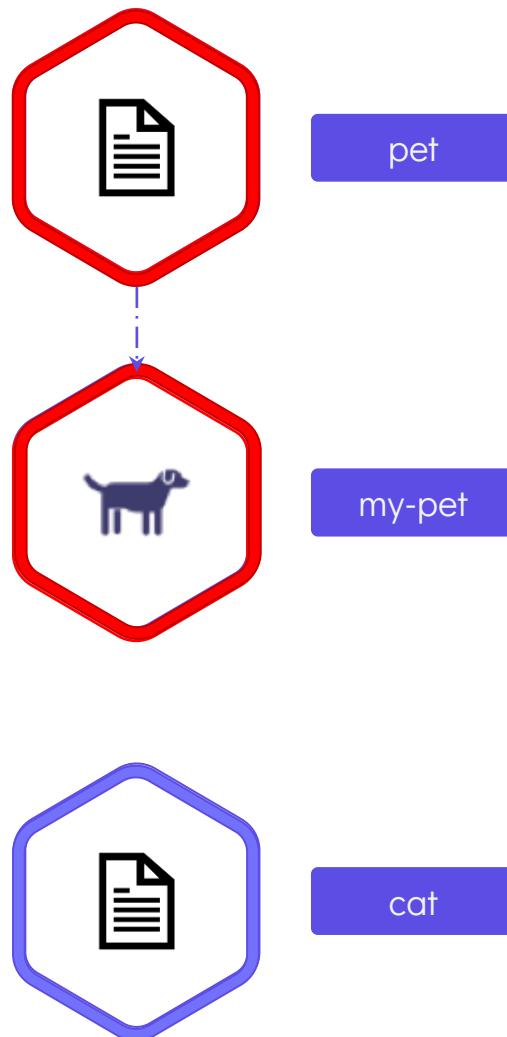
```
Plan: 0 to add, 0 to change, 2 to destroy.
```

```
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
```

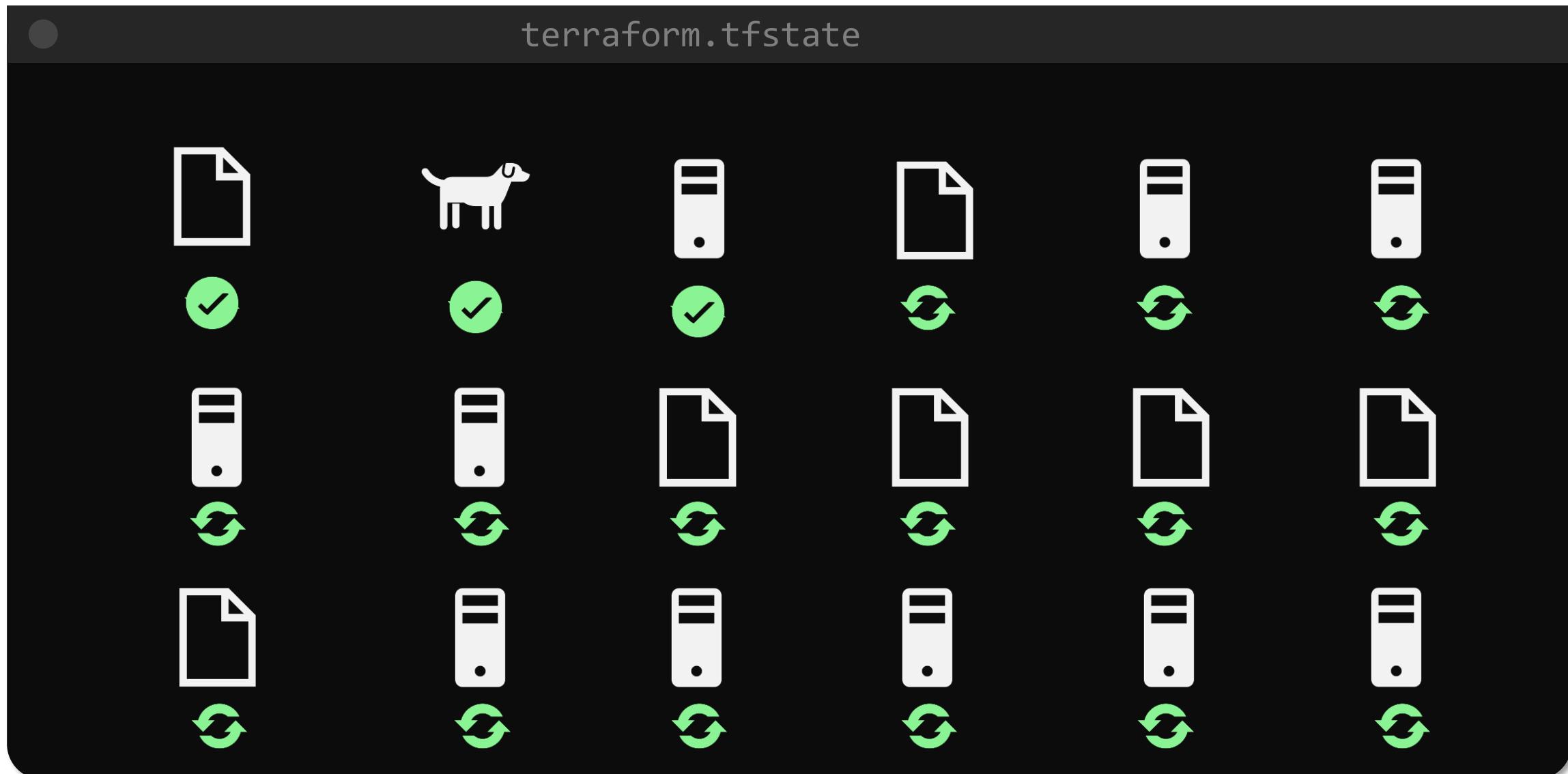
```
Enter a value: yes
```

```
local_file.pet: Destroying...
[id=28b373c6c1fa3fce132a518eadd0175c98f37f20]
local_file.pet: Destruction complete after 0s
```

```
random_pet.my-pet: Destroying... [id=yak]
random_pet.my-pet: Destruction complete after 0s
```



Performance



Performance

terraformer.tfstate

```
{  
  "version": 4,  
  "terraform_version": "0.13.0",  
  "serial": 4,  
  "lineage": "e35dde72-a943-de50-3c8b-1df8986e5a31",  
  "outputs": {},  
  "resources": [  
    {  
      "mode": "managed",  
      "type": "local_file",  
      "name": "pet",  
      "instances": [  
        {  
          "schema_version": 0,  
          "attributes": {  
            "content": "We love pets!",  
            "content_base64": null,  
            "directory_permission": "0777",  
            ...  
          }  
        }  
      ]  
    }  
  ]  
}
```

>_

```
$ terraform plan --refresh=false
```

An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

-/+ destroy and then create replacement

Terraform will perform the following actions:

```
# local_file.cat must be replaced  
-/+ resource "local_file" "pet" {  
  ~ content              = "I like cats too!" ->  
  "Dogs are awesome!" # forces replacement  
  directory_permission = "0777"  
  file_permission       = "0777"  
  filename               = "/root/pets.txt"  
  ~ id                   =  
  "cba595b7d9f94ba1107a46f3f731912d95fb3d2c" -> (known  
  after apply)  
}
```

Plan: 1 to add, 0 to change, 1 to destroy.

Collaboration

```
terraform.tfstate

{
  "version": 4,
  "terraform_version": "0.13.0",
  "serial": 4,
  "lineage": "e35dde72-a943-de50-3c8b-1df8986e5a31",
  "outputs": {},
  "resources": [
    {
      "mode": "managed",
      "type": "local_file",
      "name": "pet",
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "content": "We love pets!",
            "content_base64": null,
            "directory_permission": "0777",
            ...
        }
      ]
    }
  ]
}
```

```
>_
$ ls
main.tf variables.tf terraform.tfstate
```



AWS S3

HashiCorp Consul

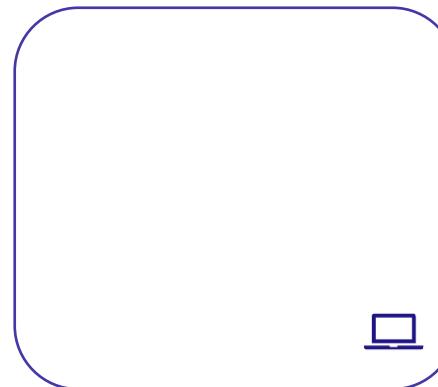
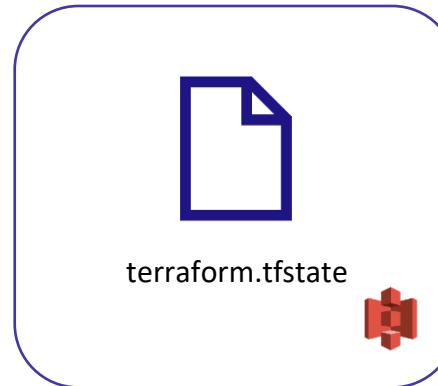
Google Cloud Storage

Terraform Cloud

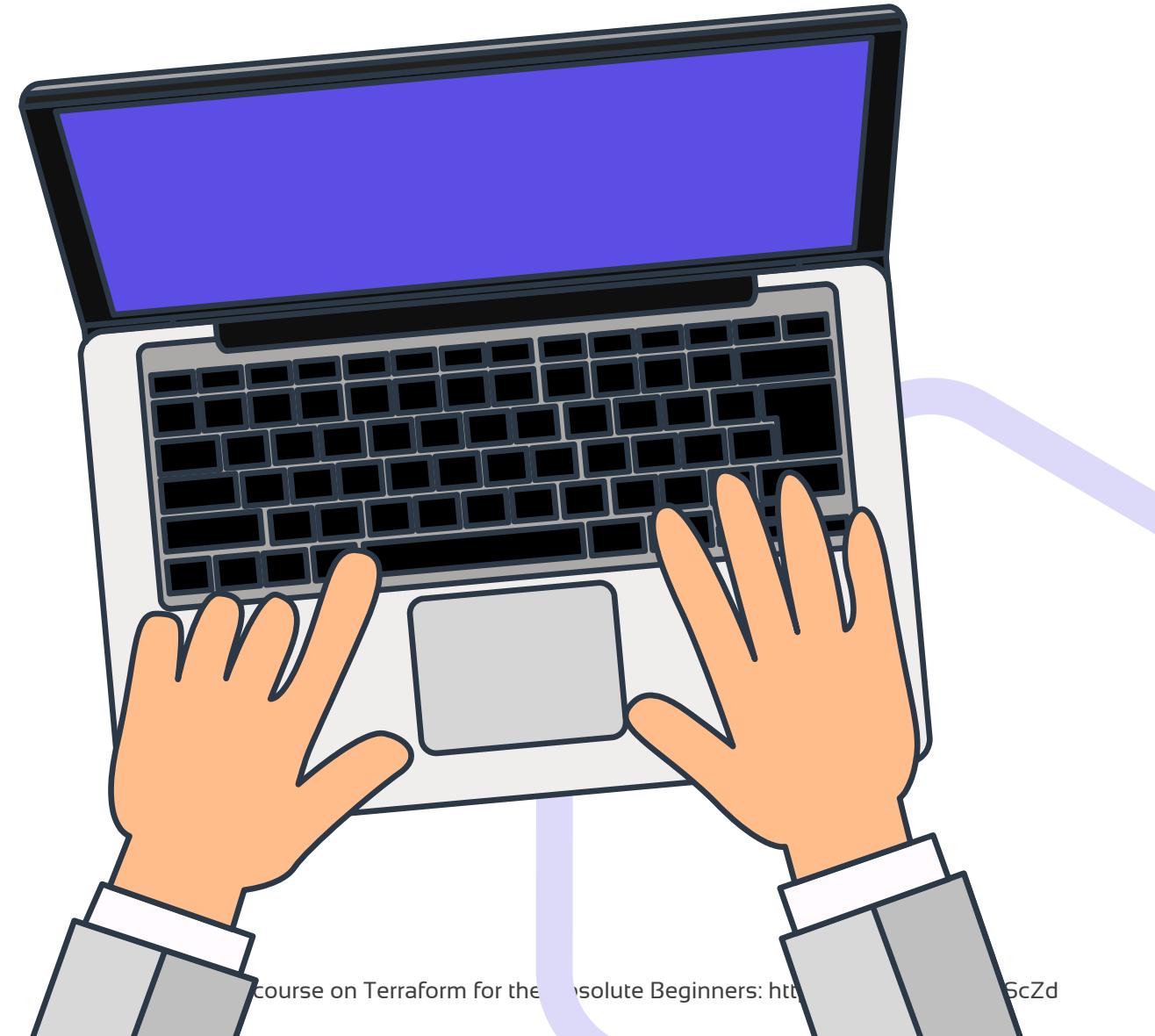
Collaboration

terraform.tfstate

```
{  
  "version": 4,  
  "terraform_version": "0.13.0",  
  "serial": 4,  
  "lineage": "e35dde72-a943-de50-3c8b-1df8986e5a31",  
  "outputs": {},  
  "resources": [  
    {  
      "mode": "managed",  
      "type": "local_file",  
      "name": "pet",  
      "instances": [  
        {  
          "schema_version": 0,  
          "attributes": {  
            "content": "We love pets!",  
            "content_base64": null,  
            "directory_permission": "0777",  
            ...  
          }  
        }  
      ]  
    }  
  ]  
}
```



HANDS-ON LABS



Terraform State Considerations

Sensitive Data

```
terraform.tfstate

{
  "mode": "managed",
  "type": "aws_instance",
  "name": "dev-ec2",
  "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
  "instances": [
    {
      "schema_version": 1,
      "attributes": {
        "ami": "ami-0a634ae95e11c6f91",
        .
        .
        .
        "primary_network_interface_id": "eni-0ccd57b1597e633e0",
        "private_dns": "ip-172-31-7-21.us-west-2.compute.internal",
        "private_ip": "172.31.7.21",
        "public_dns": "ec2-54-71-34-19.us-west-2.compute.amazonaws.com",
        "public_ip": "54.71.34.19",
        "root_block_device": [
          {
            "delete_on_termination": true,
            "device_name": "/dev/sda1",
            "encrypted": false,
            "iops": 100,
            "kms_key_id": ""
          }
        ]
      }
    }
  ]
}
```

Terraform State Considerations

Remote State Backends



terraform.tfstate

```
{  
  "mode": "managed",  
  "type": "aws_instance",  
  "name": "dev-ec2",  
  "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",  
  "instances": [  
    {  
      "schema_version": 1,  
      "attributes": {  
        "ami": "ami-0a634ae95e11c6f91",  
        ·  
        ·  
        "primary_network_interface_id": "eni-0ccd57b1597e633e0",  
        "private_dns": "ip-172-31-7-21.us-west-2.compute.internal",  
        "private_ip": "172.31.7.21",  
        "public_dns": "ec2-54-71-34-19.us-west-2.compute.amazonaws.com",  
        "public_ip": "54.71.34.19",  
        "root_block_device": [  
          {  
            "delete_on_termination": true,  
            "device_name": "/dev/sda1",  
            "encrypted": false,  
            "iops": 100,  
            "kms_key_id": "",  
            "volume_id": "vol-070720a3636979c22",  
            "volume_size": 8  
          }  
        ]  
      }  
    }  
  ]  
}
```



main.tf

```
resource "local_file" "pet" {  
  filename = "/root/pet.txt"  
  content  = "My favorite pet is Mr.Whiskers!"  
}  
resource "random_pet" "my-pet" {  
  length = 1  
}  
resource "local_file" "cat" {  
  filename = "/root/cat.txt"  
  content  = "I like cats too!"  
}
```

No Manual Edits

```
terraform.tfstate

{
  "mode": "managed",
  "type": "aws_instance",
  "name": "dev-ec2",
  "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
  "instances": [
    {
      "schema_version": 1,
      "attributes": {
        "ami": "ami-0a634ae95e11c6f91",
        .
        .
        .
        "primary_network_interface_id": "eni-0ccd57b1597e633e0",
        "private_dns": "ip-172-31-7-21.us-west-2.compute.internal",
        "private_ip": "172.31.7.21",
        "public_dns": "ec2-54-71-34-19.us-west-2.compute.amazonaws.com",
        "public_ip": "54.71.34.19",
        "root_block_device": [
          {
            "delete_on_termination": true,
            "device_name": "/dev/sda1",
            "encrypted": false,
            "iops": 100,
            "kms_key_id": ""
          }
        ]
      }
    }
  ]
}
```

Terraform Commands

terraform validate

main.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
    file_permissions = "0700"  
}
```

>_

```
$ terraform validate  
Success! The configuration is valid.  
  
$ terraform validate  
  
Error: Unsupported argument  
  
on main.tf line 4, in resource "local_file" "pet":  
  4:     file_permissions = "0777"  
  
An argument named "file_permissions" is not expected  
here. Did you mean "file_permission"?
```

terraform fmt

main.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
    file_permission = "0700"  
}
```

>_

```
$ terraform fmt  
main.tf
```

terraform fmt

main.tf

```
resource "local_file" "pet" {  
    filename      = "/root/pets.txt"  
    content       = "We love pets!"  
    file_permission = "0700"  
}
```

>_

```
$ terraform fmt  
main.tf
```

terraform show

>_

```
$ terraform show

# local_file.pet:
resource "local_file" "pet" {
    content          = "We love pets!"
    directory_permission = "0777"
    file_permission      = "0777"
    filename           = "/root/pets.txt"
    id                =
"cba595b7d9f94ba1107a46f3f731912d95fb3d2c"
}
```

>_

```
$ terraform show -json

{"format_version": "0.1", "terraform_version": "0.13.0", "values": {"root_module": {"resources": [{"address": "local_file.pet", "mode": "managed", "type": "local_file", "name": "pet", "provider_name": "registry.terraform.io/hashicorp/local", "schema_version": 0, "values": {"content": "We love pets!", "content_base64": null, "directory_permission": "0777", "file_permission": "0777", "filename": "/root/pets.txt", "id": "cba595b7d9f94ba1107a46f3f731912d95fb3d2c", "sensitive_content": null}}}]}}
```

terraform providers

main.tf

```
resource "local_file" "pet" {  
    filename      = "/root/pets.txt"  
    content       = "We love pets!"  
    file_permission = "0700"  
}
```

>_

```
$ terraform providers
```

```
Providers required by configuration:
```

```
.  
└ provider[registry.terraform.io/hashicorp/local]
```

```
Providers required by state:
```

```
provider[registry.terraform.io/hashicorp/local]
```

```
$ terraform providers mirror /root/terraform/new_local_file
```

- Mirroring hashicorp/local...
 - Selected v1.4.0 with no constraints
 - Downloading package for windows_amd64...
 - Package authenticated: signed by HashiCorp

terraform output

main.tf

```
resource "local_file" "pet" {  
    filename      = "/root/pets.txt"  
    content       = "We love pets!"  
    file_permission = "0777"  
}  
resource "random_pet" "cat" {  
    length      = "2"  
    separator   = "-"  
}  
output content {  
    value      = local_file.pet.content  
    sensitive  = false  
    description = "Print the content of the file"  
}  
output pet-name {  
    value      = random_pet.cat.id  
    sensitive  = false  
    description = "Print the name of the pet"  
}
```

>_

```
$ terraform output  
content = We love pets!  
pet-name = huge-owl
```

```
$ terraform output pet-name  
pet-name = huge-owl
```

terraform refresh

main.tf

```
resource "local_file" "pet" {  
    filename      = "/root/pets.txt"  
    content       = "We love pets!"  
    file_permission = "0777"  
}  
resource "random_pet" "cat" {  
    length      = "2"  
    separator   = "-"  
}
```

> _

\$ terraform refresh

```
random_pet.cat: Refreshing state... [id=huge-owl]  
local_file.pet: Refreshing state...  
[id=cba595b7d9f94ba1107a46f3f731912d95fb3d2c]
```

\$ terraform plan

```
Refreshing Terraform state in-memory prior to plan...  
The refreshed state will be used to calculate this  
plan, but will not be  
persisted to local or remote state storage.
```

```
random_pet.cat: Refreshing state... [id=huge-owl]  
local_file.pet: Refreshing state...  
[id=cba595b7d9f94ba1107a46f3f731912d95fb3d2c]  
-----
```

No changes. Infrastructure is up-to-date.

terraform graph

main.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content  = "My favorite pet is ${random_pet.my-pet.id}"  
}  
resource "random_pet" "my-pet" {  
    prefix = "Mr"  
    separator = "."  
    length = "1"  
}
```

> _

```
$ terraform graph  
digraph {  
    compound = "true"  
    newrank = "true"  
    subgraph "root" {  
        "[root] local_file.pet (expand)" [label =  
        "local_file.pet", shape = "box"]  
        "[root]  
provider[\"registry.terraform.io/hashicorp/local\"]" [label =  
        "provider[\"registry.terraform.io/hashicorp/local\"]", shape =  
        "diamond"]  
        "[root]  
provider[\"registry.terraform.io/hashicorp/random\"]" [label =  
        "provider[\"registry.terraform.io/hashicorp/random\"]", shape =  
        "diamond"]  
        "[root] random_pet.my-pet (expand)" [label =  
        "random_pet.my-pet", shape = "box"]  
        "[root] local_file.pet (expand)" -> "[root]  
provider[\"registry.terraform.io/hashicorp/local\"]"  
        "[root] local_file.pet (expand)" -> "[root]  
random_pet.my-pet (expand)"  
        "[root] meta.count-boundary (EachMode fixup)" ->  
        "[root] local_file.pet (expand)"  
        "[root]  
provider[\"registry.terraform.io/hashicorp/local\"] (close)" ->  
        "[root] local_file.pet (expand)"]  
}
```

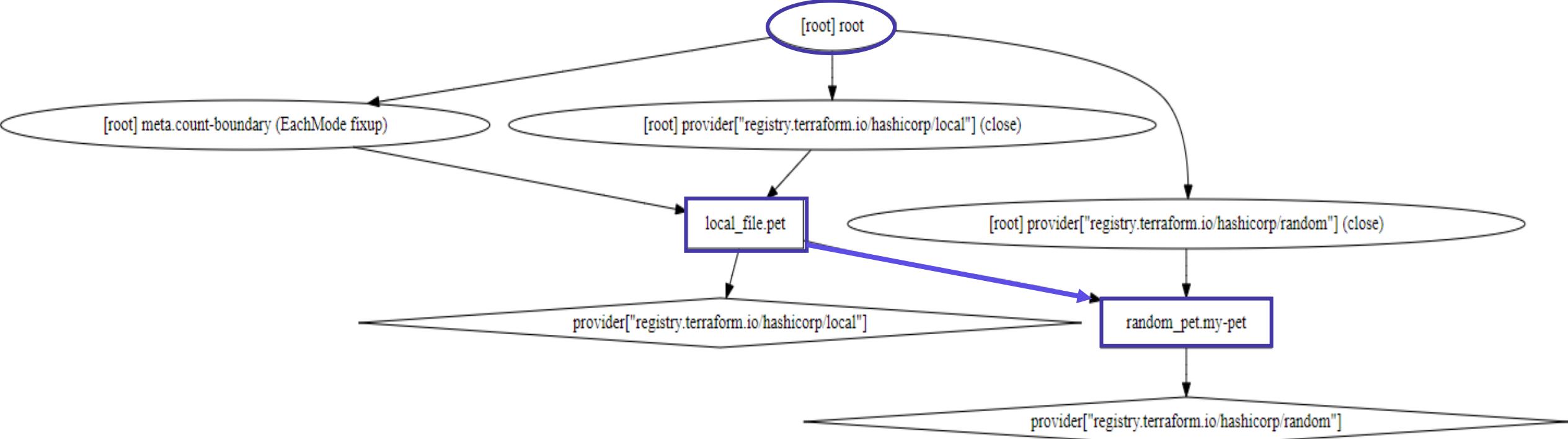
terraform graph

main.tf

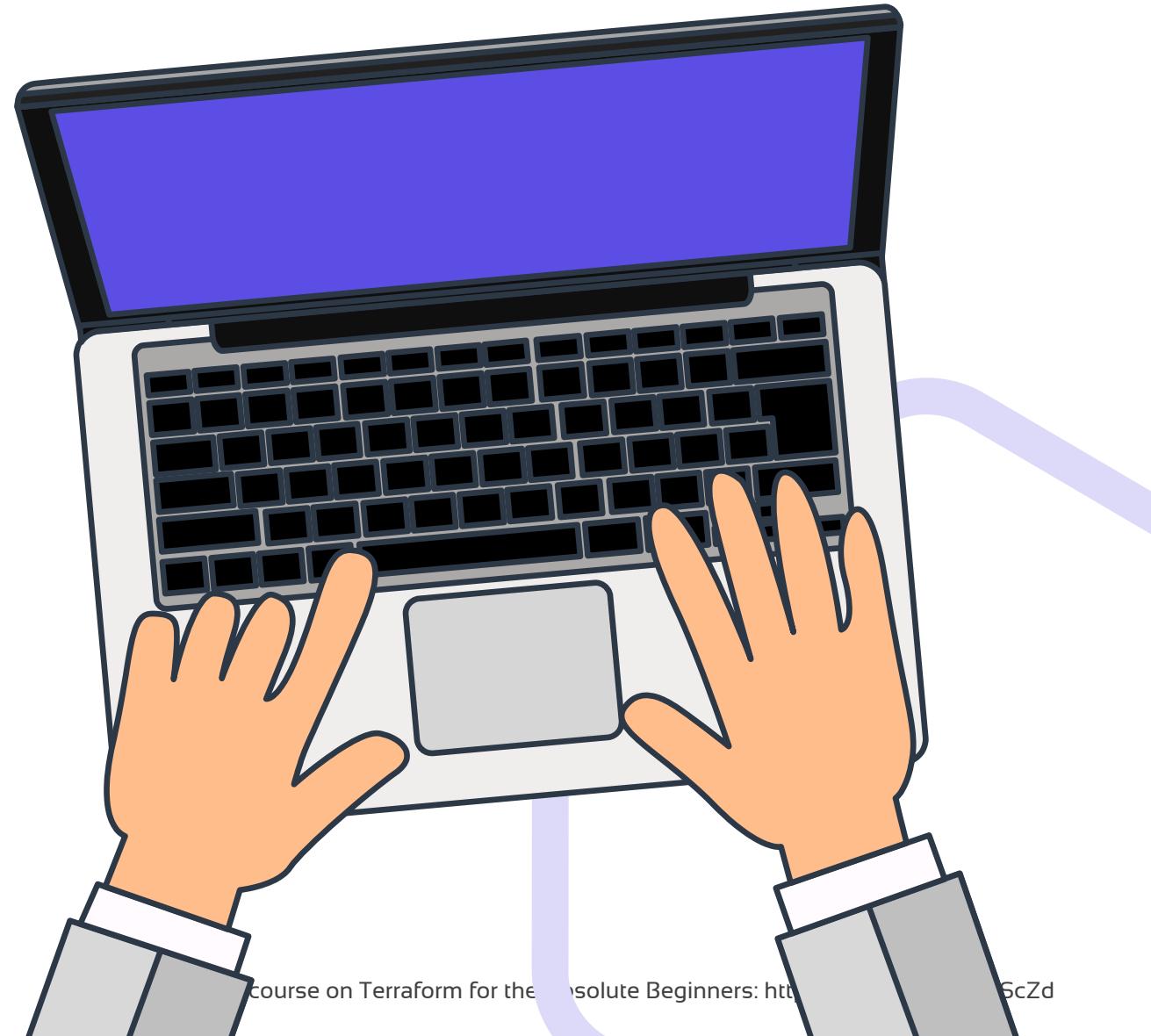
```
resource "local_file" "pet" {
  filename = "/root/pets.txt"
  content  = "My favorite pet is ${random_pet.my-pet.id}"
}
```

>_

```
$ apt update
$ apt install graphviz -y
$ terraform graph | dot -Tsvg > graph.svg
```



HANDS-ON LABS



Mutable vs Immutable Infrastructure

terraform validate

main.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
    file_permission = "0700"  
}
```



>_

```
$ terraform apply  
  
# local_file.pet must be replaced  
-/+ resource "local_file" "pet" {  
    content          = "We love pets!"  
    directory_permission = "0777"  
    ~ file_permission      = "0777" -> "0700" # forces  
replacement  
    filename          = "/root/pet.txt"  
    ~ id              =  
"5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf" -> (known after  
apply)  
}
```

Plan: 1 to add, 0 to change, 1 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

```
local_file.pet: Destroying...  
[id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]  
local_file.pet: Destruction complete after 0s  
local_file.pet: Creating...
```

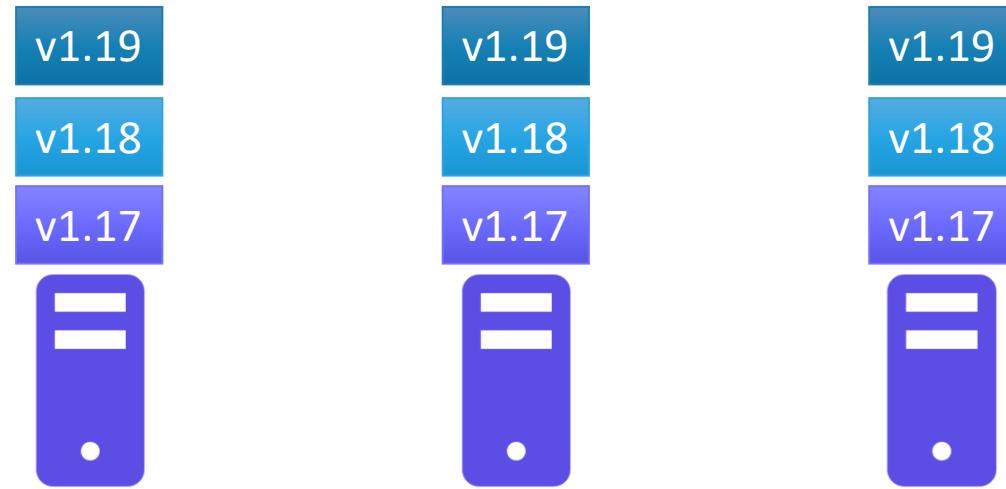


upgrade-nginx.sh

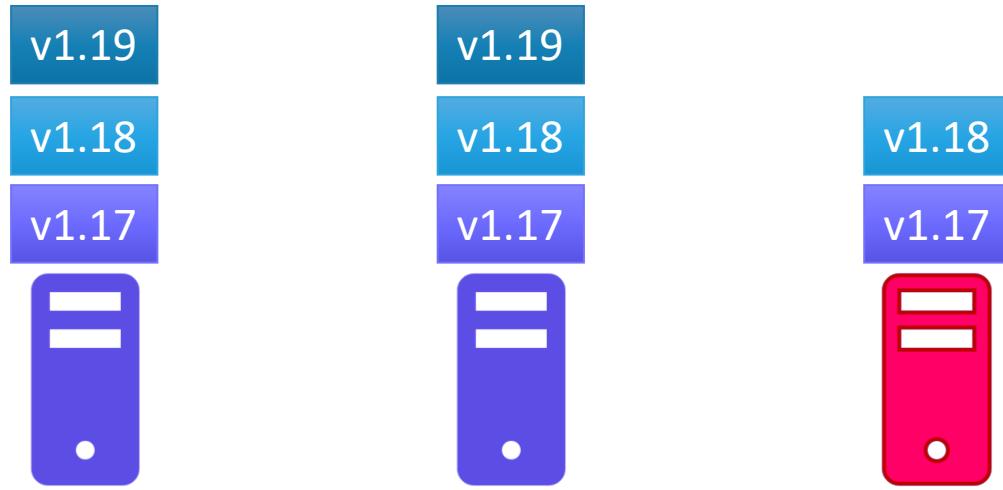


ANSIBLE

Mutable Infrastructure



Configuration Drift











Immutable Infrastructure



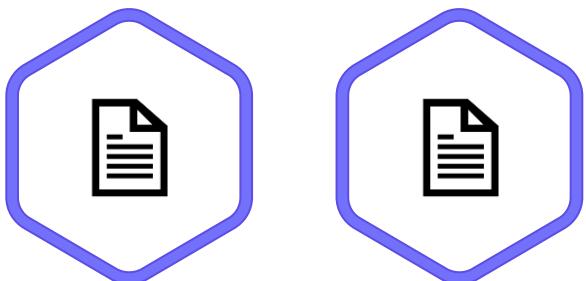
Immutable Infrastructure



Immutable Infrastructure

main.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
    file_permission = "0700"  
}
```



>_

```
$ terraform apply  
  
# local_file.pet must be replaced  
-/+ resource "local_file" "pet" {  
    content              = "We love pets!"  
    directory_permission = "0777"  
    ~ file_permission     = "0777" -> "0700" # forces  
replacement  
    filename             = "/root/pet.txt"  
    ~ id                 =  
"5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf" -> (known after  
apply)  
    }  
  
Plan: 1 to add, 0 to change, 1 to destroy.  
  
local_file.pet: Destroying...  
[id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]  
local_file.pet: Destruction complete after 0s  
local_file.pet: Creating...  
local_file.pet: Creation complete after 0s  
[id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]
```

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.



KodeKloud

Check out our full course on Terraform for the Absolute Beginners: <https://kode.wiki/3PoScZd>

Lifecycle Rules

main.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
    file_permission = "0700"  
}
```



>_

```
$ terraform apply  
  
# local_file.pet must be replaced  
-/+ resource "local_file" "pet" {  
    content              = "We love pets!"  
    directory_permission = "0777"  
    ~ file_permission     = "0777" -> "0700" # forces  
replacement  
    filename             = "/root/pet.txt"  
    ~ id                 =  
"5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf" -> (known after  
apply)  
}
```

Plan: 1 to add, 0 to change, 1 to destroy.

```
local_file.pet: Destroying...  
[id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]  
local_file.pet: Destruction complete after 0s  
local_file.pet: Creating...  
local_file.pet: Creation complete after 0s  
[id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]
```

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.

create_before_destroy

main.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
    file_permission = "0700"  
  
    lifecycle {  
        create_before_destroy = true  
    }  
}
```



>_

```
$ terraform apply  
  
# local_file.pet must be replaced  
-/+ resource "local_file" "pet" {  
    content          = "We love pets!"  
    directory_permission = "0777"  
    ~ file_permission     = "0777" -> "0755" # forces repl  
    filename         = "/root/pet.txt"  
    ~ id              =  
"5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf" -> (known after ap  
    }
```

Plan: 1 to add, 0 to change, 1 to destroy.

...

```
local_file.pet: Creating...  
local_file.pet: Creation complete after 0s  
[id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]
```

```
local_file.pet: Destroying...  
[id=5f8fb950ac60f7f23ef968097cda0a1fd3c11bdf]  
local_file.pet: Destruction complete after 0s
```

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.

prevent_destroy

main.tf

```
resource "local_file" "pet" {  
    filename = "/root/pets.txt"  
    content = "We love pets!"  
    file_permission = "0700"  
  
    lifecycle {  
        prevent_destroy = true  
    }  
}
```



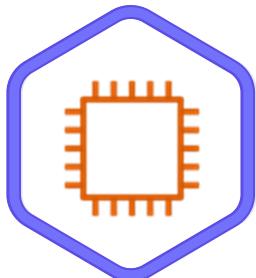
>_

```
$ terraform apply  
local_file.my-pet: Refreshing state...  
[id=cba595b7d9f94ba1107a46f3f731912d95fb3d2c]  
  
Error: Instance cannot be destroyed  
  
on main.tf line 1:  
  1: resource "local_file" "my-pet" {  
  
Resource local_file.my-pet has  
lifecycle.prevent_destroy set, but the plan calls  
for this resource to be destroyed. To avoid this error  
and continue with the plan, either disable  
lifecycle.prevent_destroy or reduce the scope of the  
plan using the -target flag.
```

ignore_changes

main.tf

```
resource "aws_instance" "webserver" {  
    ami           = "ami-0edab43b6fa892279"  
    instance_type = "t2.micro"  
    tags = {  
        Name = "ProjectA-Webserver"  
    }  
}
```



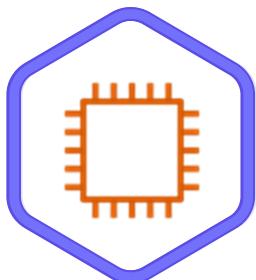
>_

```
$ terraform apply  
...  
Terraform will perform the following actions:  
  
# aws_instance.webserver will be created  
+ resource "aws_instance" "webserver" {  
    + ami           = "ami-0edab43b6fa892279"  
    + get_password_data = false  
    + host_id       = (known after apply)  
    + id            = (known after apply)  
    + instance_state = (known after apply)  
    + instance_type   = "t2.micro"  
    + tags          = {  
        + "Name"      = "ProjectA-WebServer"  
    }  
.  
aws_instance.webserver: Creation complete after 33s [id=i-  
05cd83b221911acd5]  
  
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

ignore_changes

main.tf

```
resource "aws_instance" "webserver" {
  ami           = "ami-0edab43b6fa892279"
  instance_type = "t2.micro"
  tags = {
    Name = "ProjectA-Webserver"
  }
}
```



>_

```
$ terraform apply
aws_instance.webserver: Refreshing state... [id=i-05cd83b221911acd5]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  ~ update in-place
```

Terraform will perform the following actions:

```
# aws_instance.webserver will be updated in-place
~ resource "aws_instance" "webserver" {
  .
  .
  ~ tags = {
    ~ "Name" = "ProjectB-WebServer" -> "ProjectA-WebServer"
  }
  .
  .
  Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```



ProjectB-WebServer

i-05cd83b221911acd5



Running



?

t2.micro

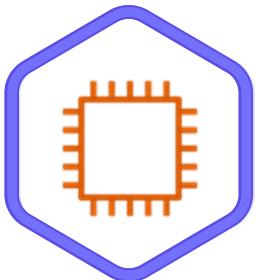


2/2 checks ...

ignore_changes

main.tf

```
resource "aws_instance" "webserver" {
    ami           = "ami-0edab43b6fa892279"
    instance_type = "t2.micro"
    tags = {
        Name = "ProjectA-Webserver"
    }
    lifecycle {
        ignore_changes = [
            tags
        ]
    }
}
```



>_

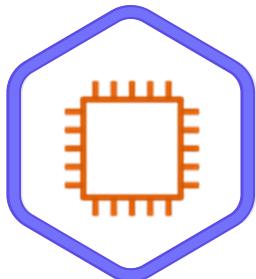
```
$ terraform apply
aws_instance.webserver: Refreshing state... [id=i-05cd83b221911acd5]

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
```

ignore_changes

main.tf

```
resource "aws_instance" "webserver" {  
    ami           = "ami-0edab43b6fa892279"  
    instance_type = "t2.micro"  
    tags = {  
        Name = "ProjectA-Webserver"  
    }  
    lifecycle {  
        ignore_changes = all  
  
    }  
}
```

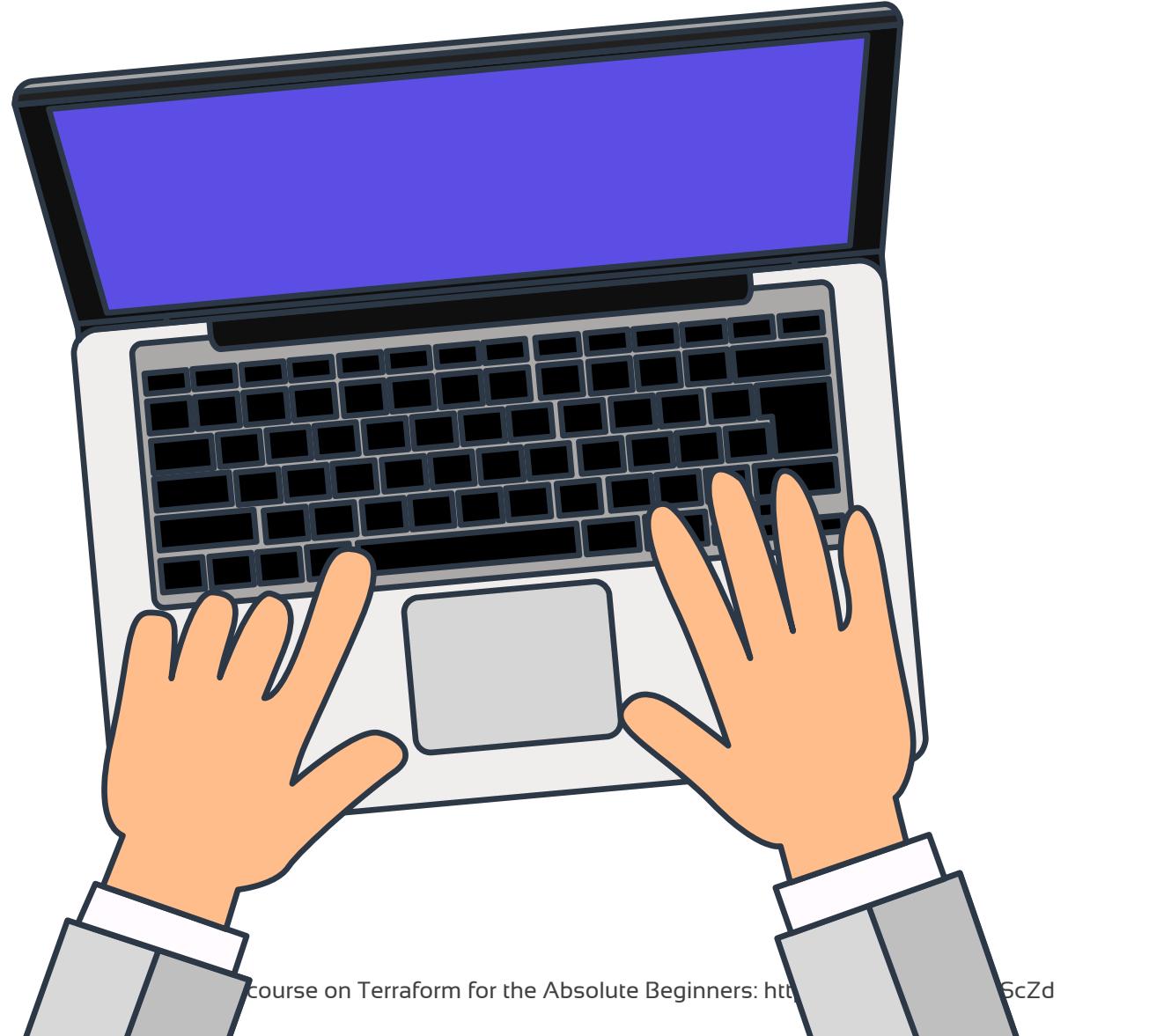


>_

```
$ terraform apply  
aws_instance.webserver: Refreshing state... [id=i-  
05cd83b221911acd5]  
  
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
```

Order	Option	
1	create_before_destroy	Create the resource first and then destroy older
2	prevent_destroy	Prevents destroy of a resource
3	ignore_changes	Ignore Changes to Resource Attributes (specific/all)

HANDS-ON LABS





KodeKloud

Check out our full course on Terraform for the Absolute Beginners: <https://kode.wiki/3PoScZd>

Data Sources



puppet

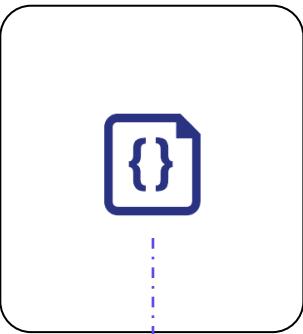


Real World Infrastructure



terraform.tfstate





```
>_
$ cat /root/dog.txt
Dogs are awesome!
```

main.tf

```
resource "local_file" "pet" {
  filename = "/root/pets.txt"
  content = "We love pets!"
}
```

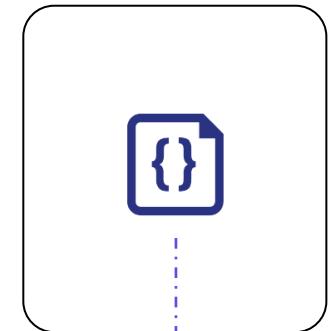
Real World Infrastructure



terraform.tfstate

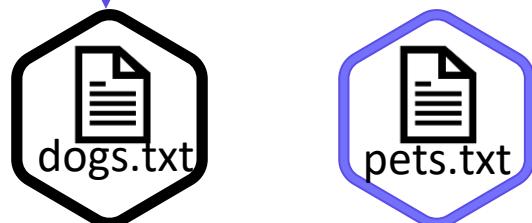


Data Sources



```
>_
$ cat /root/dog.txt
Dogs are awesome!
```

Real World Infrastructure

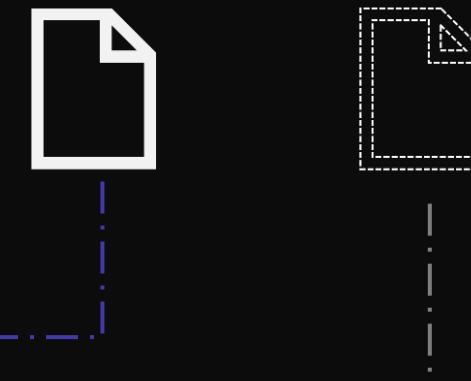


main.tf

```
resource "local_file" "pet" {
  filename = "/root/pets.txt"
  content = data.local_file.dog.content
}

data "local_file" "dog" {
  filename = "/root/dog.txt"
}
```

terraform.tfstate



LOCAL DOCUMENTATION

Filter

- local provider
- ▼ Resources
 - local_file
- ▼ Data Sources
 - local_file

Argument Reference

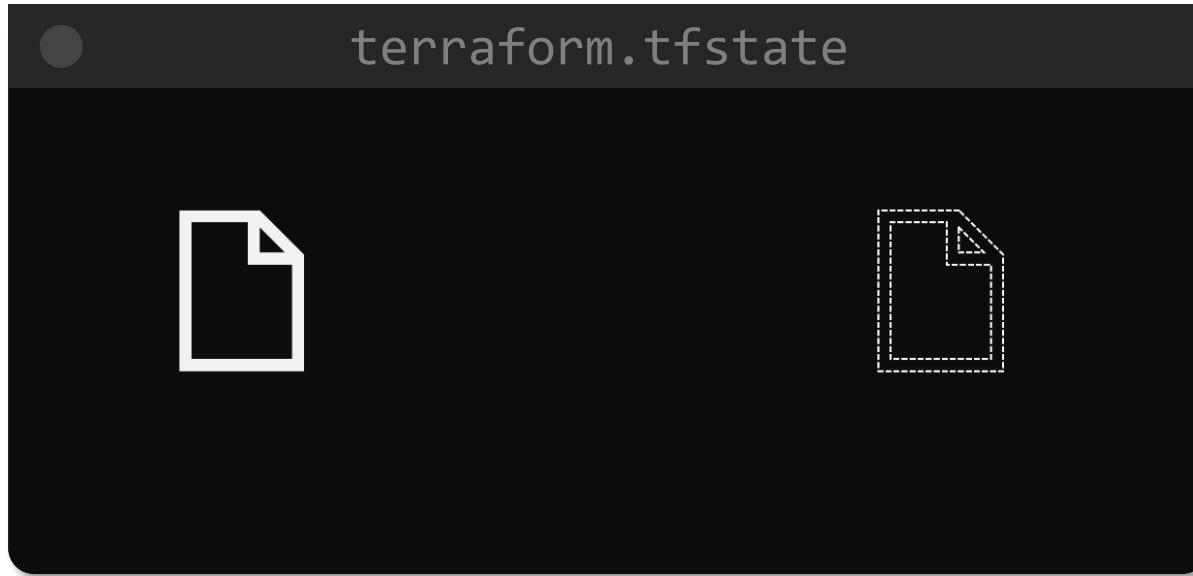
The following argument is required:

- `filename` - (Required) The path to the file that will be read. The data source will return an error if the file does not exist.

Attributes Exported

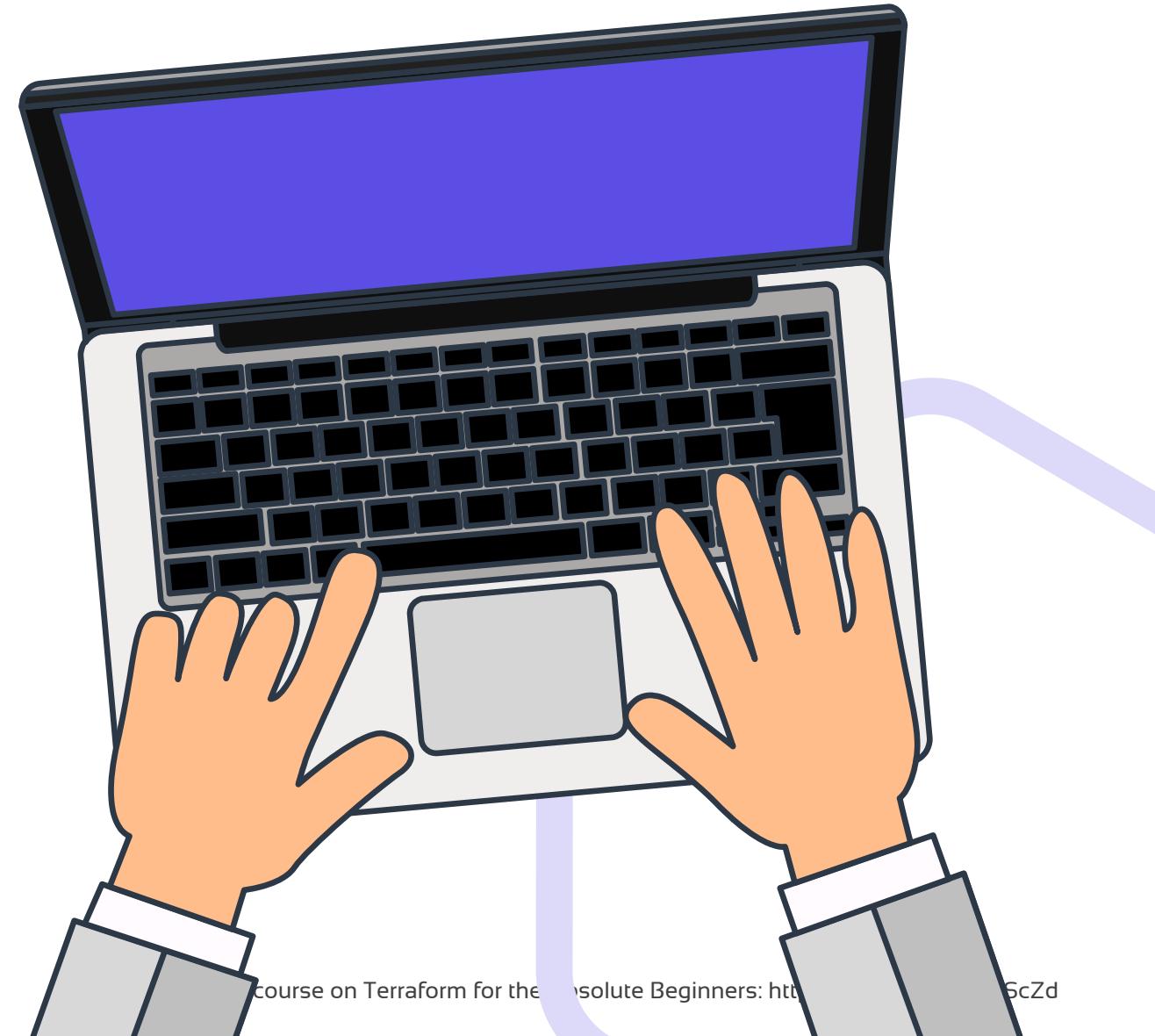
The following attribute is exported:

- `content` - The raw content of the file that was read.
- `content_base64` - The base64 encoded version of the file content (use this when dealing with binary data).



Resource	Data Source
Keyword: resource	Keyword: data
Creates, Updates, Destroys Infrastructure	Only Reads Infrastructure
Also called Managed Resources	Also called Data Resources

HANDS-ON LABS





KodeKloud

Check out our full course on Terraform for the Absolute Beginners: <https://kode.wiki/3PoScZd>

Meta Arguments

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename  
    content = var.content  
}
```

variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
}  
variable "content" {  
    default = "I love pets!"  
}
```



Shell Scripts

```
create_files.sh
```

```
#!/bin/bash

for i in {1..3}
do
    touch /root/pet${i}
done
```

```
>_
```

```
$ ls -ltr /root/
-rw-r--r-- 1 root root 0 Sep 9 02:04 pet2
-rw-r--r-- 1 root root 0 Sep 9 02:04 pet1
-rw-r--r-- 1 root root 0 Sep 9 02:04 pet3
```

Iteration	filename
1	/root/pet1
2	/root/pet2
3	/root/pet3

Meta Arguments

depends_on

```
main.tf
```

```
resource "local_file" "pet" {
  filename = var.filename
  content  = var.content
  depends_on = [
    random_pet.my-pet
  ]
}
resource "random_pet" "my-pet" {
  prefix   = var.prefix
  separator = var.separator
  length    = var.length
}
```

lifecycle

```
main.tf
```

```
resource "local_file" "pet" {
  filename = "/root/pets.txt"
  content  = "We love pets!"
  file_permission = "0700
  lifecycle  {
    create_before_destroy  = true
  }
}
```



KodeKloud

Check out our full course on Terraform for the Absolute Beginners: <https://kode.wiki/3PoScZd>

Count

count

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename  
    count    = 3  
}
```



variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
}
```

>_

```
$ terraform plan  
[Output Truncated]  
Terraform will perform the following actions:  
...  
# local_file.pet[2] will be created  
+ resource "local_file" "pet" {  
    + directory_permission = "0777"  
    + file_permission     = "0777"  
    + filename            = "/root/pets.txt"  
    + id                  = (known after apply)  
}
```

Plan: 3 to add, 0 to change, 0 to destroy.

count

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename  
    count   = 3  
}
```

pet[0]



pet[1]



pet[2]



variables.tf

```
variable "filename" {  
    default = "/root/pets.txt"  
}
```

>_

```
$ ls /root  
pet.txt
```

count

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename[count.index]  
    count   = 3  
}
```

pet[0]



pet[1]



pet[2]



variables.tf

```
variable "filename" {  
    default = [  
        "/root/pets.txt",  
        "/root/dogs.txt",  
        "/root/cats.txt"  
    ]  
}
```

>_

```
$ ls /root  
pets.txt  
dogs.txt  
cats.txt
```

Length Function

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename[count.index]  
    count   = length(var.filename)  
}
```

pet[0]



pet[1]



pet[2]



variables.tf

```
variable "filename" {  
    default = [  
        "/root/pets.txt",  
        "/root/dogs.txt",  
        "/root/cats.txt",  
        "/root/cows.txt",  
        "/root/ducks.txt"  
    ]  
}
```

```
>_  
$ ls /root  
pets.txt  
dogs.txt  
cats.txt
```

Length Function

variable	function	value
<code>fruits = ["apple", "banana", "orange"]</code>	<code>length(fruits)</code>	3
<code>cars = ["honda", "bmw", "nissan", "kia"]</code>	<code>length(cars)</code>	4
<code>colors = ["red", "purple"]</code>	<code>length(colors)</code>	2

LengthFunction

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename[count.index]  
  
    count   = length(var.filename)  
}
```

pet[0]



pet[1]



pet[2]



variables.tf

```
variable "filename" {  
    default = [  
        "/root/pets.txt",  
        "/root/dogs.txt",  
        "/root/cats.txt",  
        "/root/cows.txt",  
        "/root/ducks.txt"  
    ]  
}
```

```
>_  
$ ls /root  
pets.txt  
dogs.txt  
cats.txt
```

```
>_
```

```
$ terraform apply  
. .  
Terraform will perform the following actions:
```

```
# local_file.pet[0] will be created  
+ resource "local_file" "pet" {  
    + directory_permission = "0777"  
    + file_permission     = "0777"  
    + filename             = "/root/pets.txt"  
    + id                  = (known after apply)  
}
```

```
# local_file.pet[1] will be created  
+ resource "local_file" "pet" {  
    + directory_permission = "0777"  
    + file_permission     = "0777"  
    + filename             = "/root/dogs.txt"  
    + id                  = (known after apply)  
}
```

```
# local_file.pet[2] will be created  
+ resource "local_file" "pet" {  
    + directory_permission = "0777"  
    + file_permission     = "0777"  
    + filename             = "/root/cats.txt"  
    + id                  = (known after apply)  
}
```

```
>_
```

```
$ ls /root  
pet.txt  
dogs.txt  
cats.txt
```

main.tf

```
resource "local_file" "pet" {  
    filename = var.filename[count.index]  
  
    count    = length(var.filename)  
}
```

variables.tf

```
variable "filename" {  
    default = [  
  
        "/root/dogs.txt",  
        "/root/cats.txt"  
    ]  
}
```

pet[0]



pet[1]



pet[2]



main.tf

```
resource "local_file" "pet" {
    filename = var.filename[count.index]
    count    = length(var.filename)
}
```

pet[0]



pet[1]



pet[2]



variables.tf

```
variable "filename" {
    default = [
        "/root/dogs.txt",
        "/root/cats.txt"
    ]
}
```

> _

```
$ terraform plan
```

```
...
# local_file.pet[0] must be replaced
-/+ resource "local_file" "pet" {
    directory_permission = "0777"
    file_permission      = "0777"
    ~ filename            = "/root/pets.txt" -> "/root/dogs.txt" #
forces replacement
}
# local_file.pet[1] must be replaced
-/+ resource "local_file" "pet" {
    directory_permission = "0777"
    file_permission      = "0777"
    ~ filename            = "/root/dogs.txt" -> "/root/cats.txt" #
forces replacement
}
# local_file.pet[2] will be destroyed
- resource "local_file" "pet" {
    - directory_permission = "0777" -> null
    - file_permission      = "0777" -> null
}
```

main.tf

```
resource "local_file" "pet" {
  filename = var.filename[count.index]
  count    = length(var.filename)
}

output "pets" {
  value = local_file.pet
}
```

pet[0]



pet[1]



pet[2]

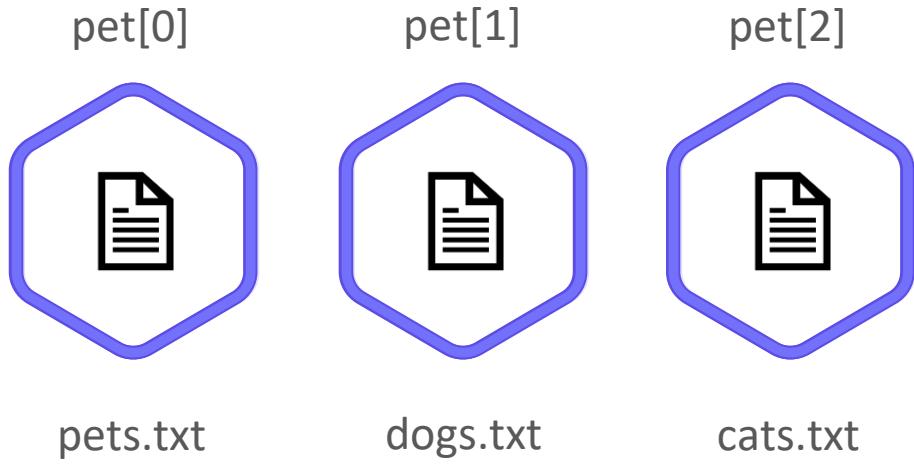


>_

```
$ terraform output
```

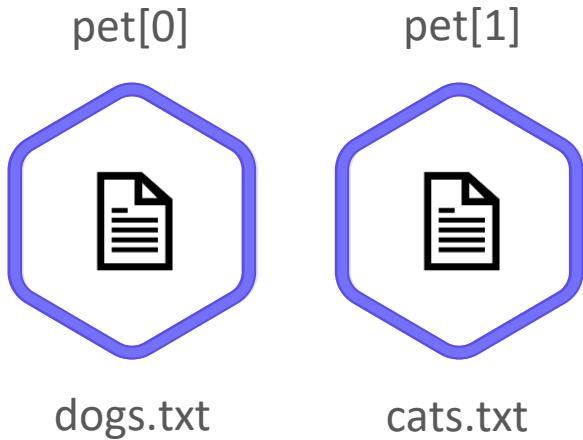
Outputs:

```
pets = [
  {
    "directory_permission" = "0777"
    "file_permission" = "0777"
    "filename" = "/root/pets.txt"
    "id" = "da39a3ee5e6b4b0d3255bfef95601890afd80709"
  },
  {
    "directory_permission" = "0777"
    "file_permission" = "0777"
    "filename" = "/root/dogs.txt"
    "id" = "da39a3ee5e6b4b0d3255bfef95601890afd80709"
  },
  {
    "directory_permission" = "0777"
    "file_permission" = "0777"
    "filename" = "/root/cats.txt"
    "id" = "da39a3ee5e6b4b0d3255bfef95601890afd80709"
  },
]
```



```
variables.tf
```

```
variable "filename" {  
    default = [  
        "/root/dogs.txt",  
        "/root/cats.txt"  
    ]  
}
```

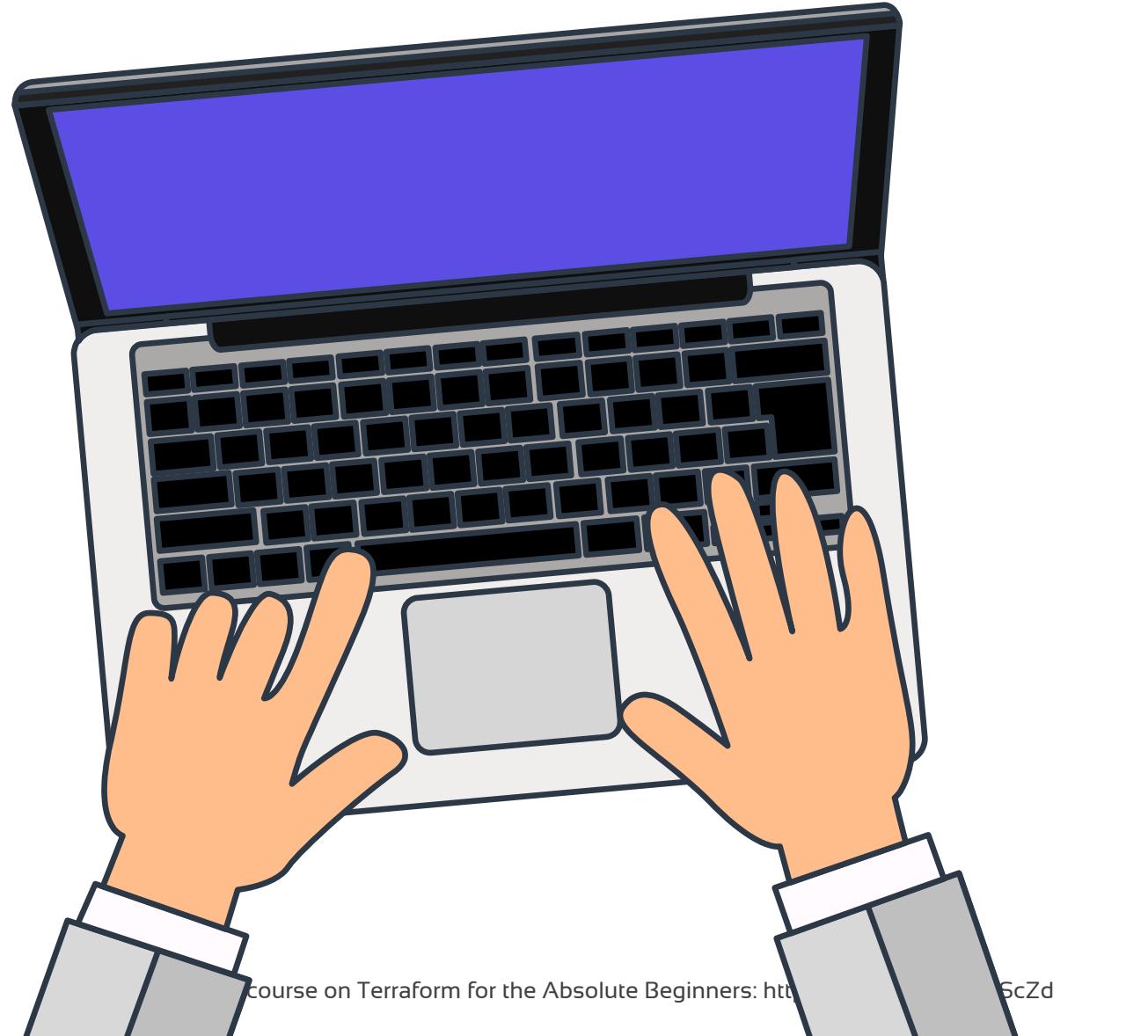


```
variables.tf
```

```
variable "filename" {
  default = [
    "/root/dogs.txt",
    "/root/cats.txt"
  ]
}
```

Resource	Resource Updates	Action
pet[0]	/root/pets.txt" -> "/root/dogs.txt"	Destroy and Replace
pet[1]	" /root/dogs.txt" -> "/root/cats.txt"	Destroy and Replace
pet[2]	Does not Exist	Destroy

HANDS-ON LABS





KodeKloud

Check out our full course on Terraform for the Absolute Beginners: <https://kode.wiki/3PoScZd>

for_each

for_each

main.tf

```
resource "local_file" "pet" {  
    filename = each.value  
    for_each = var.filename  
}
```

variables.tf

```
variable "filename" {  
    type = list(string)  
    default = [  
        "/root/pets.txt",  
        "/root/dogs.txt",  
        "/root/cats.txt"  
    ]  
}
```

pet[0]



pet[1]



pet[2]



>_

\$ terraform plan

Error: Invalid for_each argument

on main.tf line 2, in resource "local_file" "pet":
2: for_each = var.filename

The given "for_each" argument value is unsuitable: the "for_each" argument must be a map, or set of strings, and you have provided a value of type list of string.

for_each

main.tf

```
resource "local_file" "pet" {  
    filename = each.value  
    for_each = var.filename  
}
```

variables.tf

```
variable "filename" {  
    type=set(string)  
    default = [  
        "/root/pets.txt",  
        "/root/dogs.txt",  
        "/root/cats.txt"  
    ]  
}
```

pet[0]



pet[1]



pet[2]



>_

\$ terraform plan

```
Terraform will perform the following actions:  
# local_file.pet["/root/cats.txt"] will be created  
+ resource "local_file" "pet" {  
    + directory_permission = "0777"  
    + file_permission     = "0777"  
    + filename            = "/root/cats.txt"  
}  
... <output trimmed>  
Plan: 3 to add, 0 to change, 0 to destroy.
```

for_each

main.tf

```
resource "local_file" "pet" {
    filename = each.value
    [for_each = toset(var.filename)]
}
```

variables.tf

```
variable "filename" {
    type=set(string)
    default = [
        "/root/pets.txt",
        "/root/dogs.txt",
        "/root/cats.txt"
    ]
}
```

pet[0]



pet[1]



pet[2]



>_

\$ terraform plan

```
Terraform will perform the following actions:
# local_file.pet["/root/cats.txt"] will be created
+ resource "local_file" "pet" {
    + directory_permission = "0777"
    + file_permission      = "0777"
    + filename              = "/root/cats.txt"
}
... <output trimmed>
Plan: 3 to add, 0 to change, 0 to destroy.
```

for_each

main.tf

```
resource "local_file" "pet" {
  filename = each.value
  for_each = toset(var.filename)
}

output "pets" {
  value = local_file.pet
}
```

pet[0]



pet[1]



pet[2]



variables.tf

```
variable "filename" {
  type=list(string)
  default = [
    "/root/dogs.txt",
    "/root/cats.txt"
  ]
}
```

>

\$ terraform plan

Terraform will perform the following actions:

```
# local_file.pet["/root/pets.txt"] will be destroyed
+ resource "local_file" "pet" {
  + directory_permission = "0777"
  + file_permission      = "0777"
  + filename              = "/root/pets.txt"
}
... <output trimmed>
Plan: 0 to add, 0 to change, 1 to destroy.
```

for_each

main.tf

```
resource "local_file" "pet" {
  filename = each.value
  for_each = toset(var.filename)
}

output "pets" {
  value = local_file.pet
}
```

pet[0]



pet[1]



pet[2]



>_

```
$ terraform output
pets = {
  "/root/cats.txt" = {
    "directory_permission" = "0777"
    "file_permission" = "0777"
    "filename" = "/root/cats.txt"
    "id" = "da39a3ee5e6b4b0d3255bfef95601890afd80709"
  }

  "/root/dogs.txt" = {
    "directory_permission" = "0777"
    "file_permission" = "0777"
    "filename" = "/root/dogs.txt"
    "id" = "da39a3ee5e6b4b0d3255bfef95601890afd80709"
  }
}
```

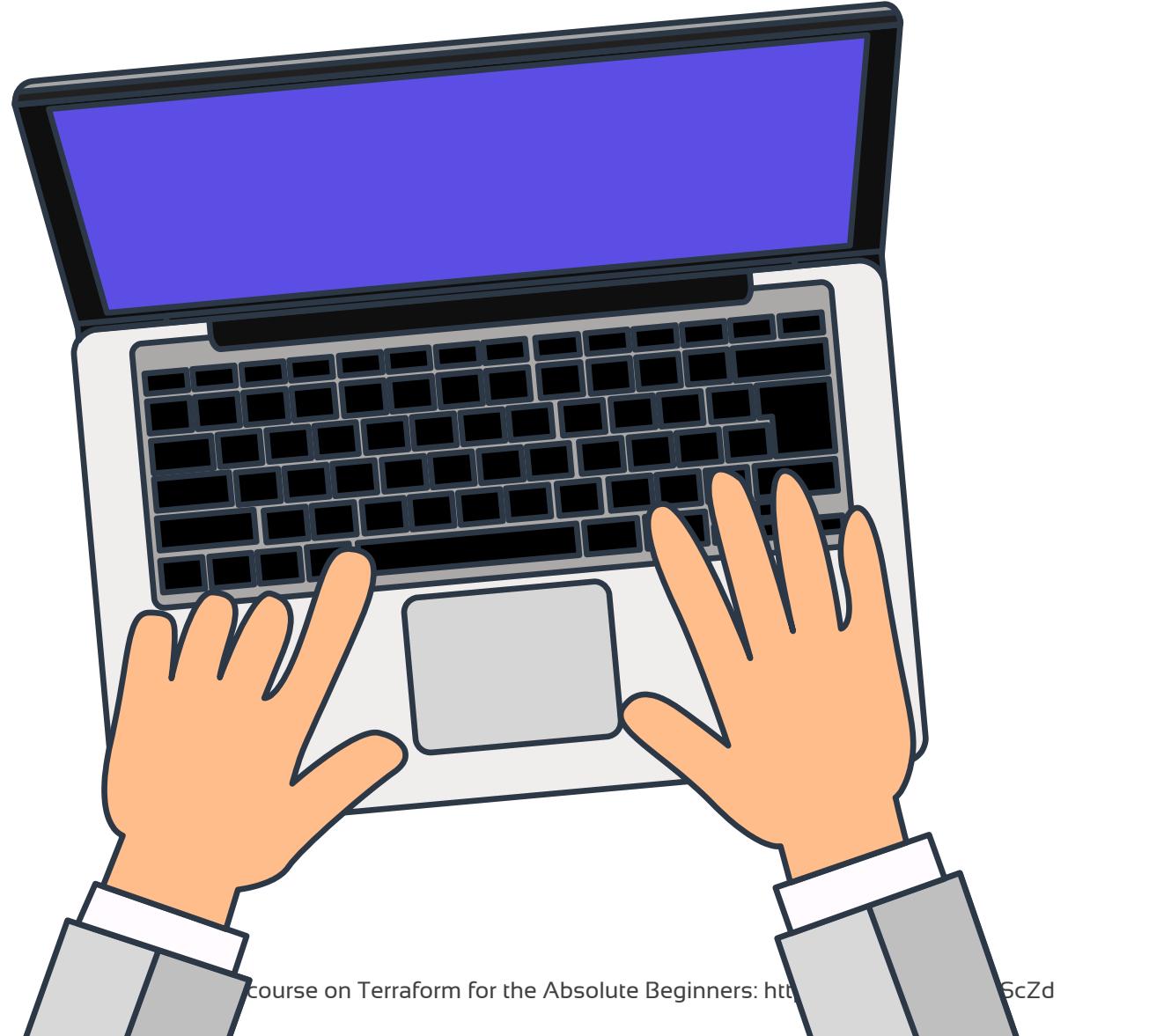
count

```
>_
$ terraform output
pets = [
  {
    "directory_permission" = "0777"
    "file_permission" = "0777"
    "filename" = "/root/pets.txt"
    "id" = "da39a3ee5e6b4b0d3255bfef95601890afd80709"
  },
  {
    "directory_permission" = "0777"
    "file_permission" = "0777"
    "filename" = "/root/dogs.txt"
    "id" = "da39a3ee5e6b4b0d3255bfef95601890afd80709"
  },
  {
    "directory_permission" = "0777"
    "file_permission" = "0777"
    "filename" = "/root/cats.txt"
    "id" = "da39a3ee5e6b4b0d3255bfef95601890afd80709"
  },
]
```

for_each

```
>_
$ terraform output
pets = {
  "/root/cats.txt" = {
    "directory_permission" = "0777"
    "file_permission" = "0777"
    "filename" = "/root/cats.txt"
    "id" = "da39a3ee5e6b4b0d3255bfef95601890afd80709"
  }
  "/root/dogs.txt" = {
    "directory_permission" = "0777"
    "file_permission" = "0777"
    "filename" = "/root/dogs.txt"
    "id" = "da39a3ee5e6b4b0d3255bfef95601890afd80709"
  }
}
```

HANDS-ON LABS





KodeKloud

Check out our full course on Terraform for the Absolute Beginners: <https://kode.wiki/3PoScZd>

Version Constraints

main.tf

```
resource "local_file" "pet" {  
    filename      = "/root/pet.txt"  
    content      = "We love pets!"  
}
```

>_

```
$ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/local...
- Installing hashicorp/local v1.4.0...
- Installed hashicorp/local v1.4.0 (signed by HashiCorp)

The following providers do not have any version constraints
in configuration, so the latest version was installed.

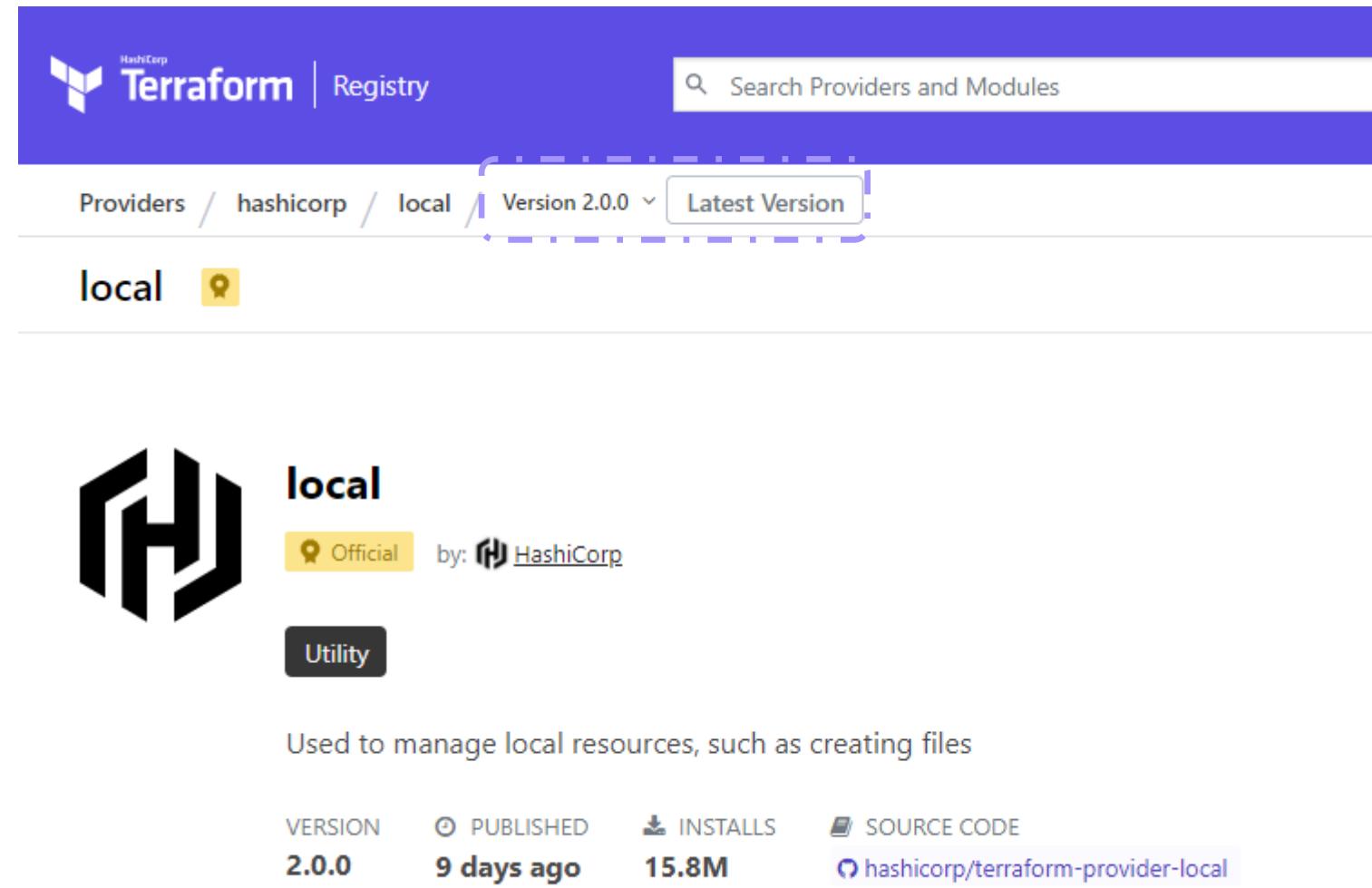
To prevent automatic upgrades to new major versions that may
contain breaking
changes, we recommend adding version constraints in a
required_providers block
in your configuration, with the constraint strings suggested
below.

```
* hashicorp/local: version = "~> 1.4.0"
```

Terraform has been successfully initialized!

```
main.tf

resource "local_file" "pet" {
    filename      = "/root/pet.txt"
    content       = "We love pets!"
}
```



The screenshot shows the Terraform Registry interface. At the top, there's a navigation bar with the HashiCorp logo, the word "Terraform", and "Registry". A search bar is on the right. Below the navigation, a breadcrumb trail shows "Providers / hashicorp / local". A dropdown menu indicates "Version 2.0.0" and "Latest Version". The main content area displays the "local" provider details. It features the HashiCorp logo, the provider name "local" in large bold letters, and a "Utility" tag. A yellow "Official" badge is present. The provider is attributed to "by: HashiCorp". A description states "Used to manage local resources, such as creating files". Below this, there are metrics: "VERSION 2.0.0", "PUBLISHED 9 days ago", "INSTALLS 15.8M", and a "SOURCE CODE" link.

main.tf

```
resource "local_file" "pet" {
    filename      = "/root/pet.txt"
    content       = "We love pets!"
}
```

Terraform | Registry

Providers / hashicorp / local Version 2.0.0 Latest Version

local

local

Official by: HashiCorp

Utility

Used to manage local resources, such as creating files

VERSION	PUBLISHED	INSTALLS	SOURCE CODE
2.0.0	9 days ago	15.8M	hashicorp/terraform-provider-local

main.tf

```
resource "local_file" "pet" {  
    filename      = "/root/pet.txt"  
    content      = "We love pets!"  
}
```

The screenshot shows the HashiCorp Terraform Registry interface. At the top, there's a navigation bar with the Terraform logo and the word "Registry". A search bar is located at the top right. Below the navigation, a breadcrumb trail shows the path: Providers / hashicorp / local / Version 2.0.0 / Latest Version. The "Latest Version" tab is selected. On the left, there's a sidebar with the "local" provider logo, which is a stylized 'H'. Below the logo, it says "local" and "Utility". To the right of the sidebar, a box titled "LATEST VERSION" displays "Version 2.0.0" with a checkmark, followed by the text "Published 9 days ago". A vertical dashed line separates this from a list of previous versions: "Version 1.4.0" (published a year ago), "Version 1.3.0" (published a year ago), "Version 1.2.2" (published a year ago), and "Version 1.2.1" (published a year ago). The "Version 2.0.0" entry is highlighted with a yellow background.

main.tf

```
resource "local_file" "pet" {
  filename      = "/root/pet.txt"
  content      = "We love pets!"
}
```

[Overview](#)[Documentation](#)[USE PROVIDER ▾](#)

How to use this provider

To install this provider, copy and paste this code into your Terraform configuration. Then, run `terraform init`.

Terraform 0.13 [Latest](#)

```
terraform {
  required_providers {
    local = {
      source = "hashicorp/local"
      version = "1.4.0"
    }
  }
}
```

```
terraform {  
  required_providers {  
    local = {  
      source = "hashicorp/local"  
      version = "1.4.0"  
    }  
  }  
  
  resource "local_file" "pet" {  
    filename      = "/root/pet.txt"  
    content      = "We love pets!"  
  }  
}
```

How to use this provider

To install this provider, copy and paste this code into your Terraform configuration. Then, run `terraform init`.

Terraform 0.13 Latest

```
terraform {  
  required_providers {  
    local = {  
      source = "hashicorp/local"  
      version = "1.4.0"  
    }  
  }  
}
```

main.tf

```
terraform {  
    required_providers {  
        local = {  
            source = "hashicorp/local"  
            version = "1.4.0"  
        }  
    }  
  
    resource "local_file" "pet" {  
        filename      = "/root/pet.txt"  
        content      = "We love pets!"  
    }  
}
```

>_

```
$ terraform init  
Initializing the backend...  
  
Initializing provider plugins...  
- Finding hashicorp/local versions matching "1.4.0"...  
- Installing hashicorp/local v1.4.0...  
- Installed hashicorp/local v1.4.0 (signed by HashiCorp)  
  
Terraform has been successfully initialized!  
  
You may now begin working with Terraform. Try running  
"terraform plan" to see  
any changes that are required for your infrastructure. All  
Terraform commands  
should now work.  
  
If you ever set or change modules or backend configuration for  
Terraform,  
rerun this command to reinitialize your working directory. If  
you forget, other  
commands will detect it and remind you to do so if necessary.
```

main.tf

```
terraform {
  required_providers {
    local = {
      source = "hashicorp/local"
      version = "> 1.2.0, < 2.0.0, != 1.4.0"
    }
  }

  resource "local_file" "pet" {
    filename      = "/root/pet.txt"
    content      = "We love pets!"
  }
}
```

>_

```
$ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Finding hashicorp/local versions matching "> 1.2.0, < 2.0.0, != 1.4.0"...
- Installing hashicorp/local v1.3.0...
- Installed hashicorp/local v1.3.0 (signed by HashiCorp)

Terraform has been successfully initialized!

main.tf

```
terraform {  
    required_providers {  
        local = {  
            source = "hashicorp/local"  
            version = "~> 1.2.0"  
        }  
    }  
  
    resource "local_file" "pet" {  
        filename      = "/root/pet.txt"  
        content      = "We love pets!"  
    }  
}
```

>_

```
$ terraform init
```

Initializing

Initializin

- Finding h

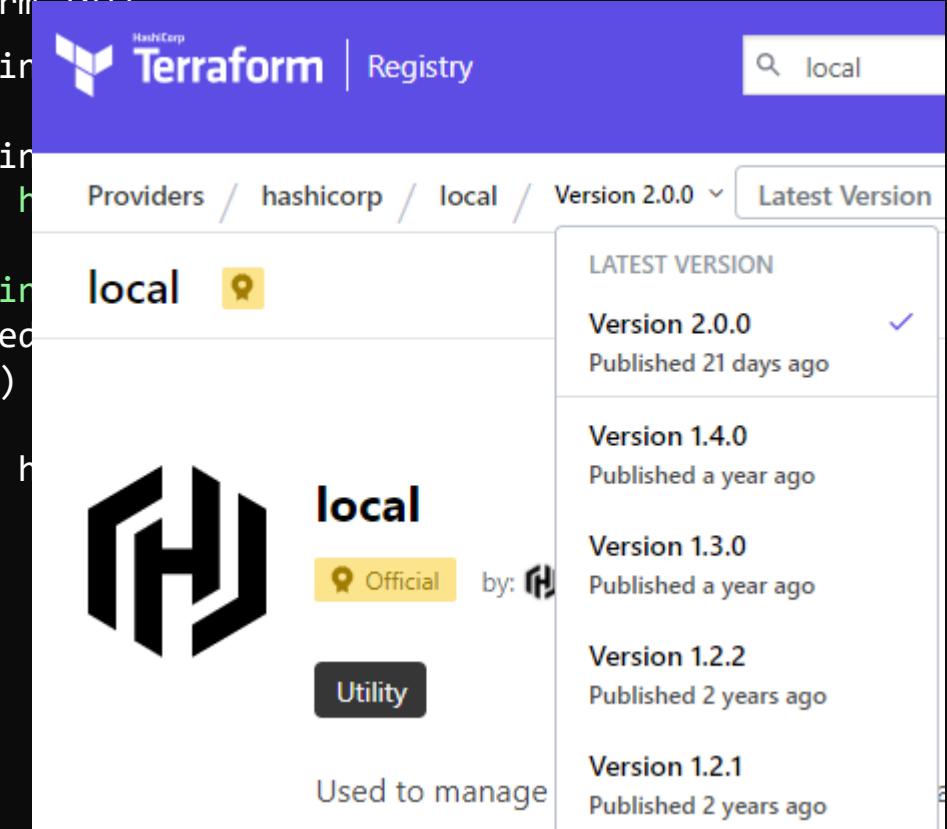
1.2.0"...

- Installin

- Installed

HashiCorp)

Terraform h



The screenshot shows the Terraform Registry interface. At the top, there's a search bar with the text 'local'. Below it, the URL path is shown: Providers / hashicorp / local / Version 2.0.0 / Latest Version. A sidebar on the left lists versions: LATEST VERSION (Version 2.0.0, Published 21 days ago), Version 1.4.0 (Published a year ago), Version 1.3.0 (Published a year ago), Version 1.2.2 (Published 2 years ago), and Version 1.2.1 (Published 2 years ago). The main content area displays the 'local' provider details. It features the HashiCorp logo, the provider name 'local' in large bold letters, a yellow 'Official' badge with a lock icon, and a 'Utility' badge. Below this, it says 'Used to manage'. To the right, there's a 'by: HashiCorp' link.



KodeKloud

Check out our full course on Terraform for the Absolute Beginners: <https://kode.wiki/3PoScZd>

Certified Entry-Level Python Programmer Certification (PCEP)

Basic
Concepts

Data Types

Operators

Flow Control

Data
Collections

Functions



ENTRY

PCEP-30-XX
Certified Entry-Level Python
Programmer Certification



ASSOCIATE

PCAP-31-XX
Certified Associate in Python
Programming Certification



PROFESSIONAL

PCPP-32-1XX
Certified Professional in Python
Programming 1 Certification



PCPP-32-2XX
Certified Professional in Python
Programming 2 Certification



Available via **OpenEDG Testing Service**

Available via **Authorized Pearson VUE Testing Centers / OnVUE online proctoring**



Introduction

Easy and intuitive programming language

Free and Open Source

Can be widely used for a variety of tasks





bash

```
$ python3
```



python3

```
$ python3
Python 3.8.2 (default, Oct  2 2020, 10:45:41)
[Clang 12.0.0 (clang-1200.0.32.27)] on darwin
Type "help", "copyright", "credits" or "license" for more information

>>>
```

Python PSF Docs PyPI Jobs Community

 python™

Search GO Socialize

About Downloads Documentation Community Success Stories News Events

Download the latest version for Windows

[Download Python 3.7.0](#)

Looking for Python with a different OS? [Python for Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#)

Looking for Python 2.7? See below for specific releases



Join the official **Python Developers Survey 2018** and win valuable prizes: [Start the survey!](#)

Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more
Python 3.5.6	2018-06-02	 Download Release Notes
Python 3.4.9	2018-06-02	 Download Release Notes
Python 3.7.0	2018-06-27	 Download Release Notes



python3

```
$ python3
Python 3.8.2 (default, Oct  2 2020, 10:45:41)
[Clang 12.0.0 (clang-1200.0.32.27)] on darwin
Type "help", "copyright", "credits" or "license" for more information

>>>
```



python3

```
$ python3
Python 3.8.2 (default, Oct  2 2020, 10:45:41)
[Clang 12.0.0 (clang-1200.0.32.27)] on darwin
Type "help", "copyright", "credits" or "license" for more information
>>> print  ( "Hello future Python programmer!" )
```



python3

```
$ python3
Python 3.8.2 (default, Oct 2 2020, 10:45:41)
[Clang 12.0.0 (clang-1200.0.32.27)] on darwin
Type "help", "copyright", "credits" or "license" for more information

>>> print ("Hello future Python programmer!")
Hello future Python programmer!
```

myfile.py

```
1 print ( "Hello future Python programmer!" )
```



bash

```
$
```

myfile.py

```
1 print ( "Hello future Python programmer!" )
```



bash

```
$ python myfile.py
```

myfile.py

```
1 print ( "Hello future Python programmer!" )
```



bash

```
$ python myfile.py  
Hello future Python programmer!  
$
```

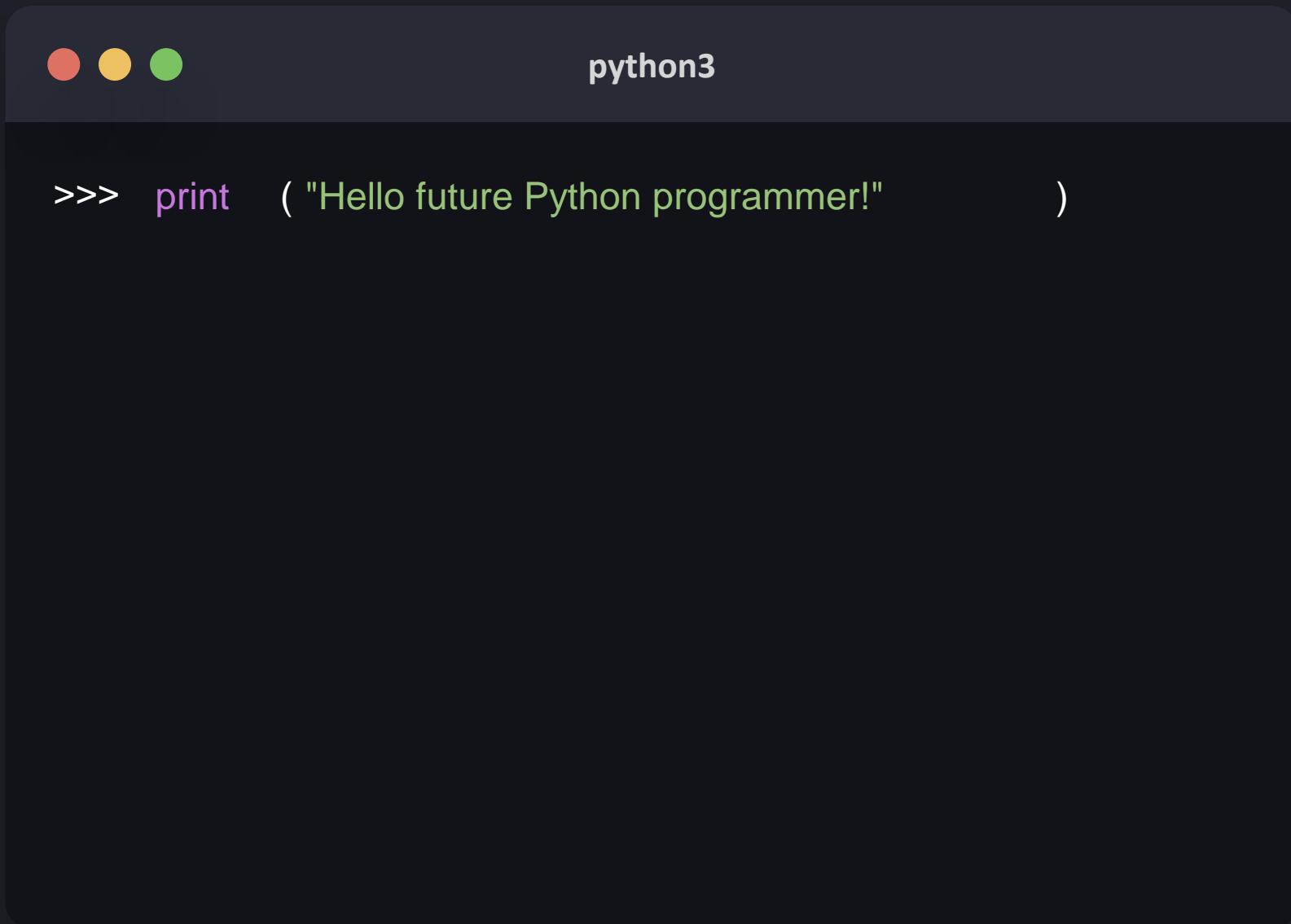


KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>

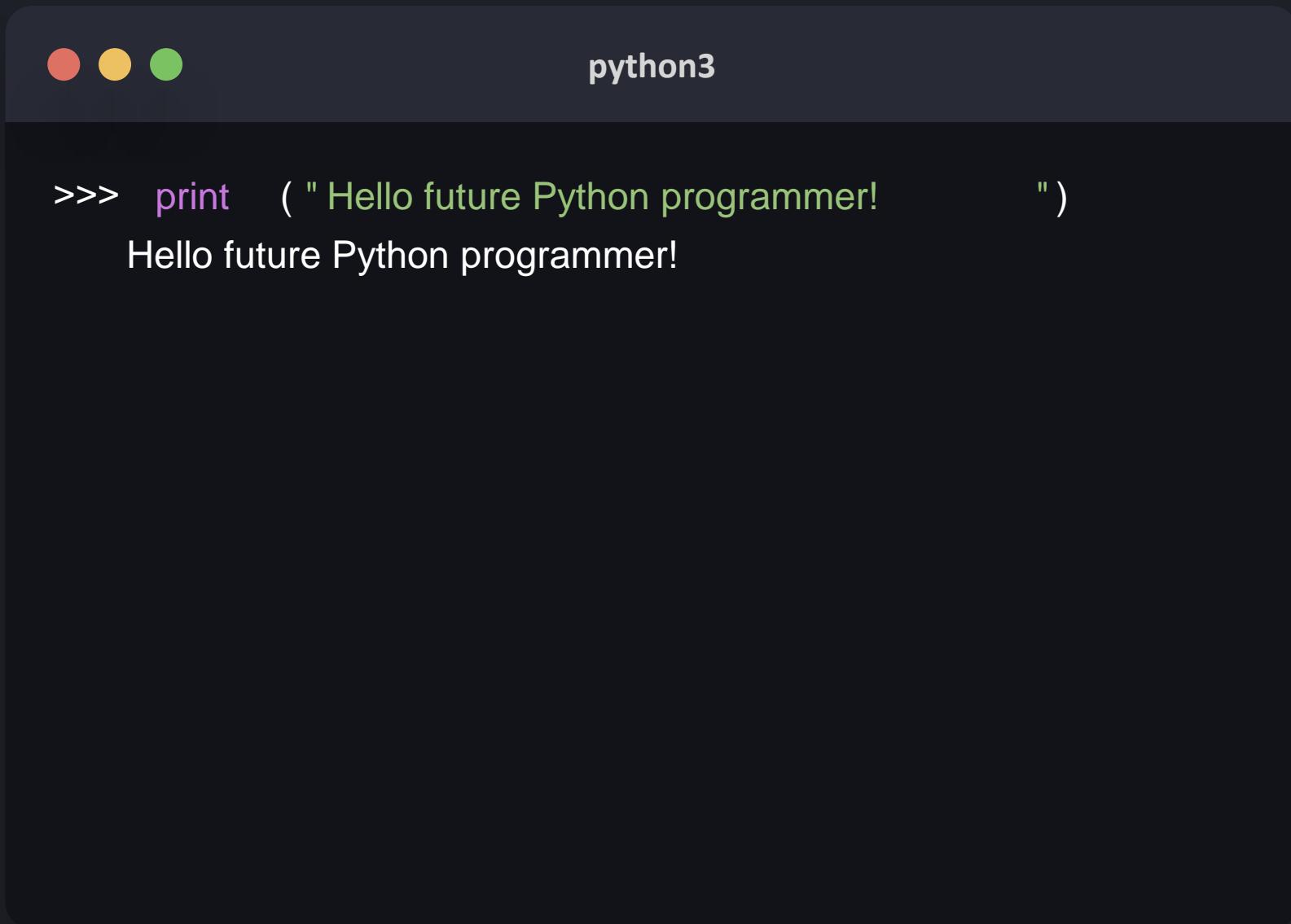


Functions - Print()

A screenshot of a macOS terminal window. The window title is "python3". In the terminal, the following Python code is displayed:

```
>>> print ("Hello future Python programmer!")
```

The text "Hello future Python programmer!" is printed in green.

A screenshot of a dark-themed terminal window. The title bar is labeled "python3". In the main pane, the following Python code is displayed:

```
>>> print( "Hello future Python programmer!" )
```

The output of the code is:

Hello future Python programmer!



python3

>>>

Hello future Python programmer!

print

(

"

Hello future Python programmer!

"

)

```
python3

>>>
Hello future Python programmer!
```

print (" Hello future Python programmer! ")



python3

```
>>> print ( "Hello future Python programmer!" )  
Hello future Python programmer!
```

Functions

A part of your code that's used to cause an effect or evaluate a value.



python3

```
>>> print ( "Hello future Python programmer!" )  
Hello future Python programmer!
```

Functions

A part of your code that's used to cause an effect or evaluate a value.

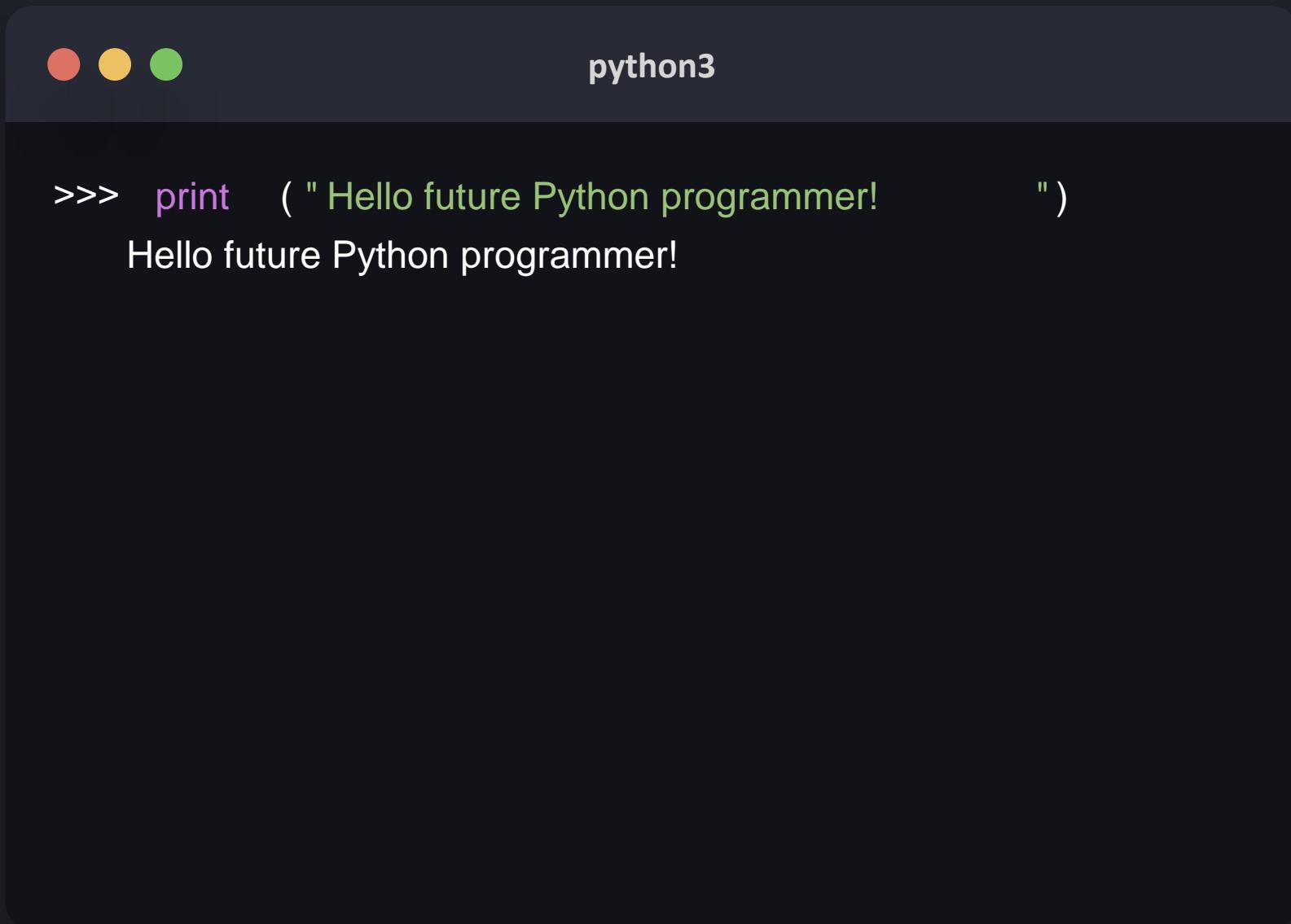
Can come from:

- Python (built-in functions)
- Modules
- Your own code



python3

```
>>> print ( "Hello future Python programmer!" )  
Hello future Python programmer!
```

A screenshot of a dark-themed terminal window. The title bar is labeled "python3". In the main pane, the following Python code is displayed:

```
>>> print( "Hello future Python programmer!" )
```

The output of the code is:

Hello future Python programmer!





print

```
( "Hello future Python programmer!" )
```



python3

>>>

Hello future Python programmer!

```
print
```

```
( "Hello future Python programmer!" )
```



```
python3
```

```
>>>
```

```
Hello future Python programmer!
```





python3

```
>>> print ( "Hello future Python programmer!" )
```

```
Hello future Python programmer!
```

```
>>> print("Python is a great language")
```

```
Python is a great language
```

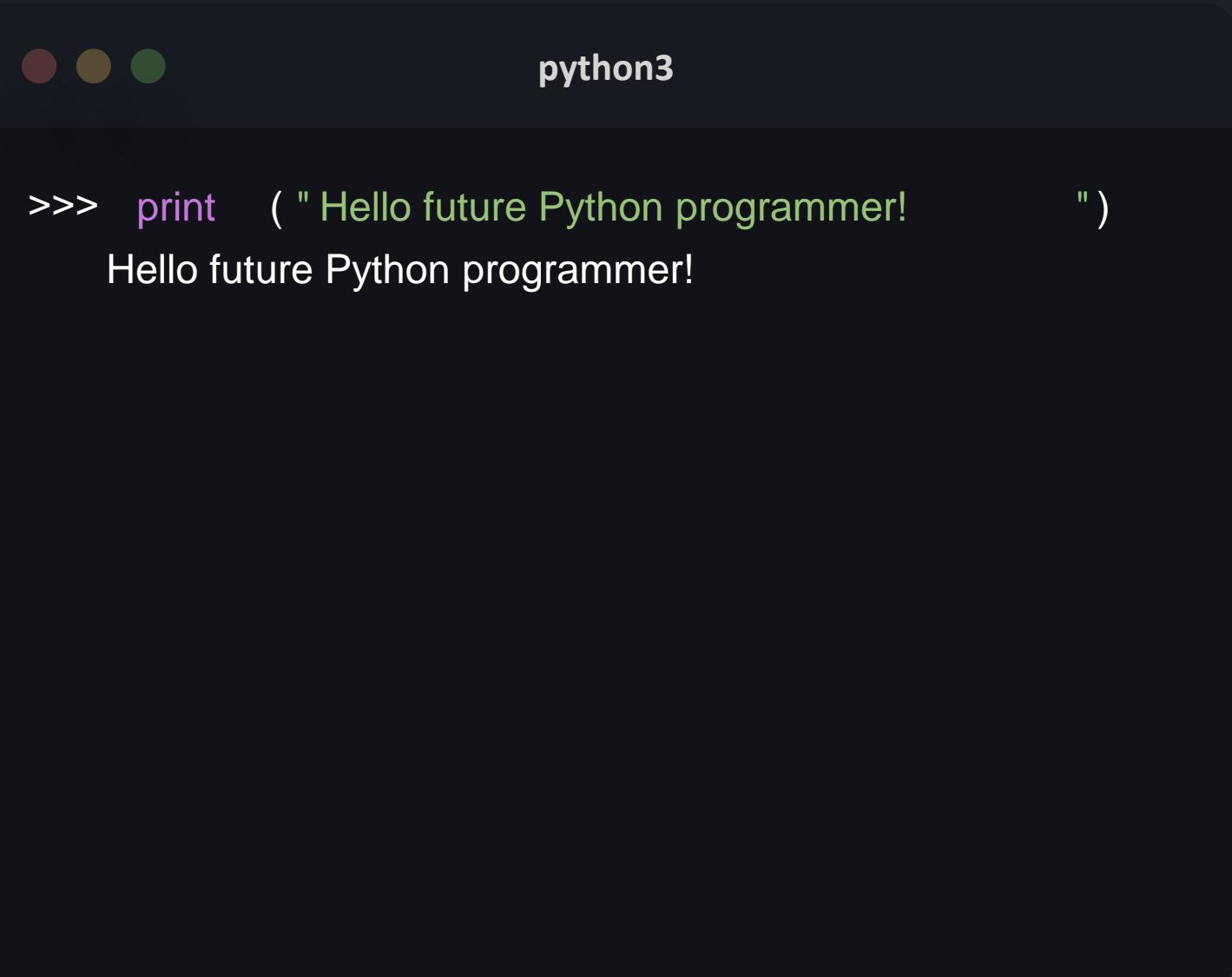
```
>>> print("Strings don't get executed as code")
```

```
Strings don't get executed as code
```

Function Execution

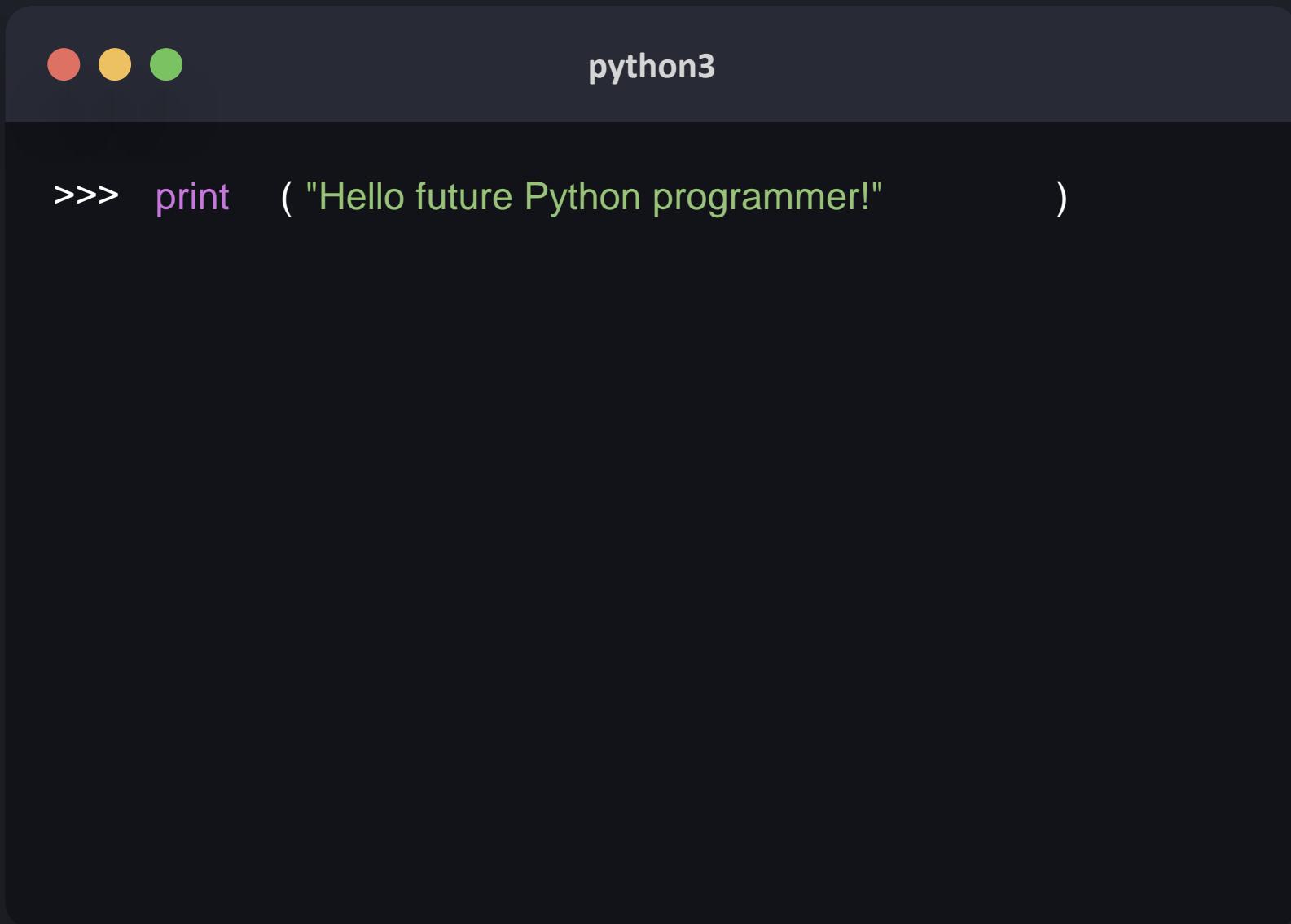
Python:

1. Checks function name
2. Checks arguments passed
3. Jumps into the function
4. Executes the function
5. Returns to your code
6. Resumes execution



A screenshot of a terminal window titled "python3". The window has three colored window control buttons at the top left. The title bar on the right says "python3". The terminal content shows a single line of code: "print ("Hello future Python programmer!")". The output of the code, "Hello future Python programmer!", is displayed below the code line.

```
>>> print ( "Hello future Python programmer!" )
Hello future Python programmer!
```

A screenshot of a macOS terminal window. The window title is "python3". In the terminal, the following Python code is displayed:

```
>>> print ("Hello future Python programmer!")
```

The text "Hello future Python programmer!" is rendered in green, indicating it is a string literal.



python3

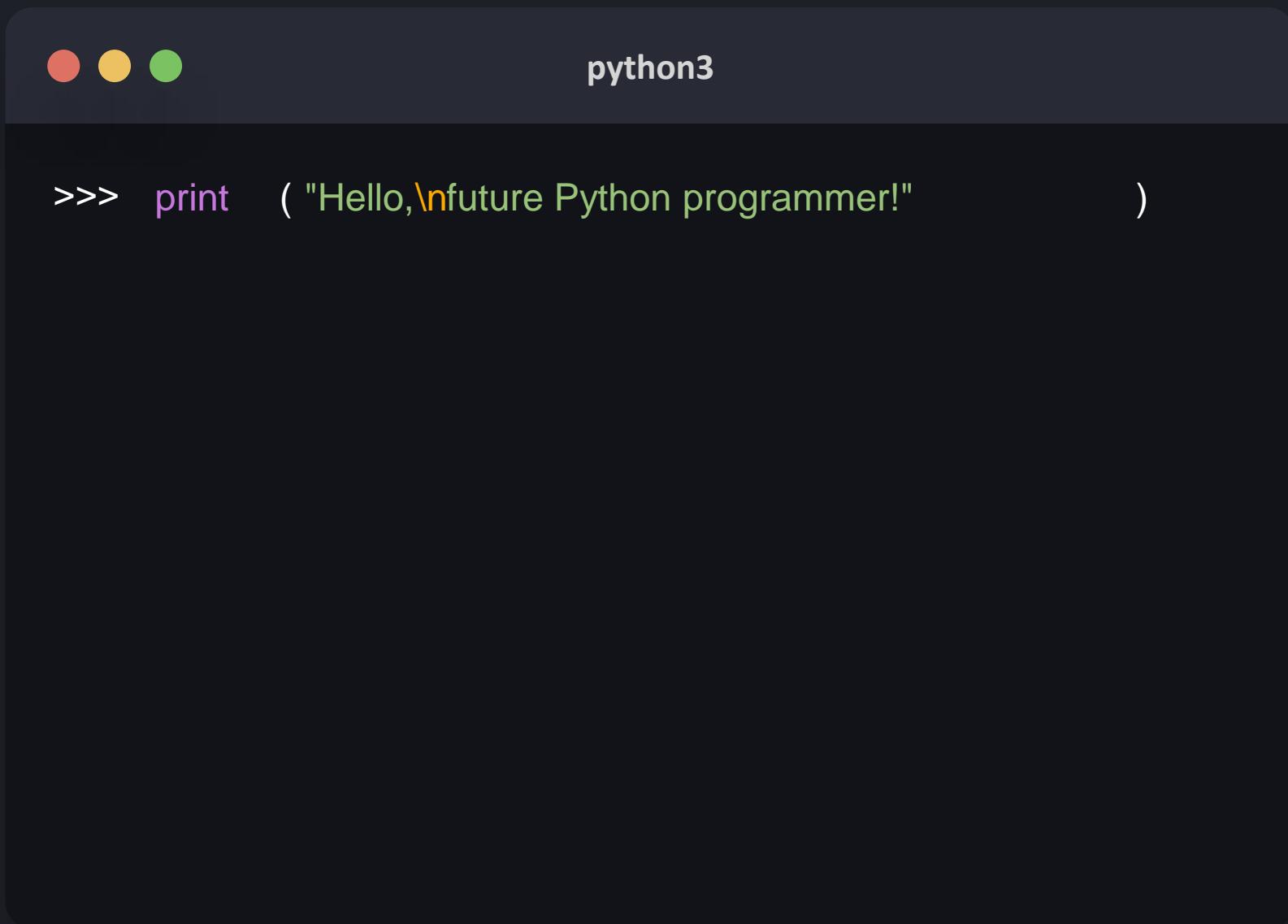
```
>>> print ( "Hello future Python programmer!" )  
print("Python is a great language")  
print("Strings don't get executed as code")
```



python3

```
>>> print ( "Hello future Python programmer!" )  
print("Python is a great language")  
print("Strings don't get executed as code")
```

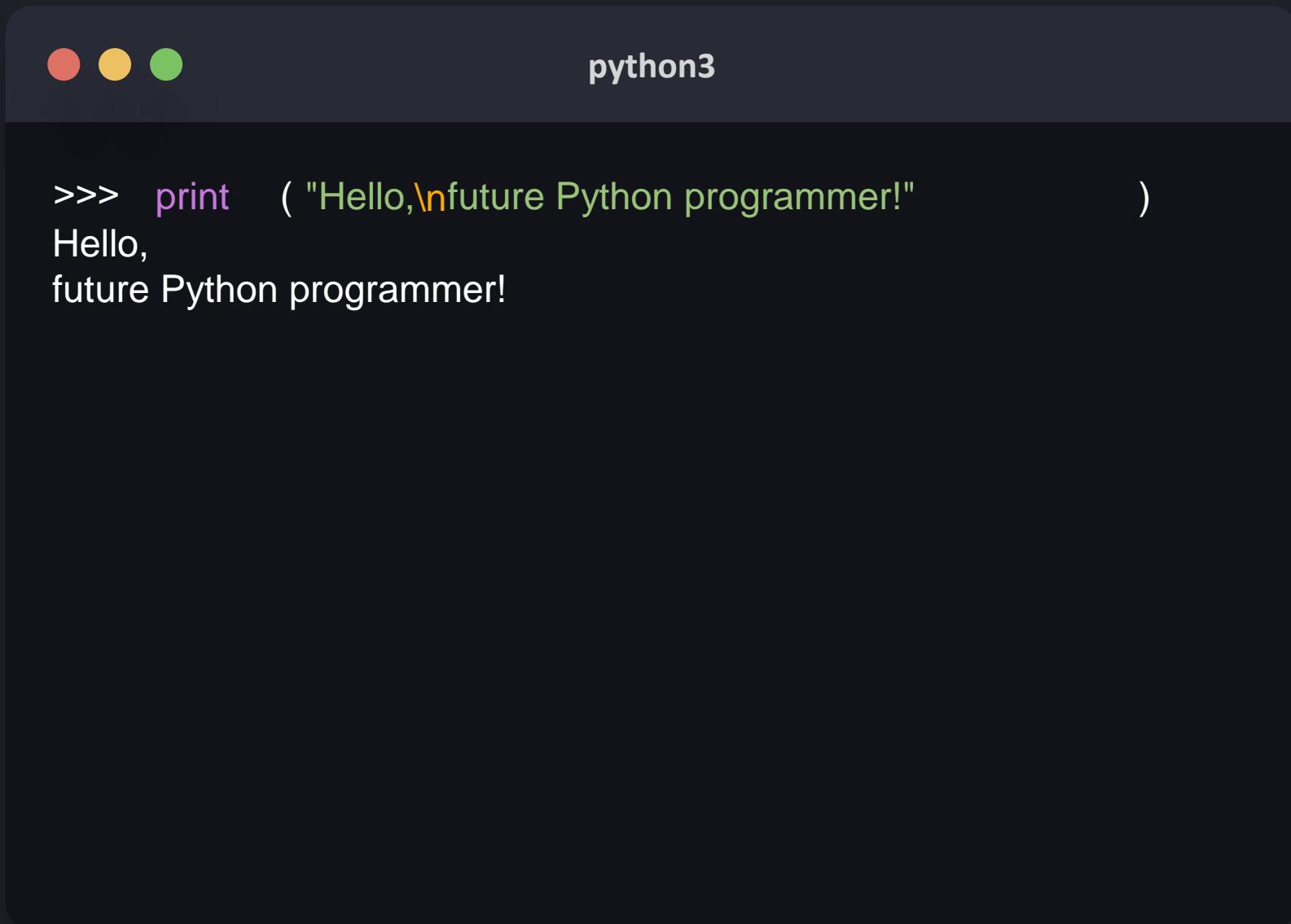
```
Hello future Python programmer!  
Python is a great language  
Strings don't get executed as code
```

A screenshot of a macOS terminal window. The window title is "python3". In the terminal, the following Python code is run:

```
>>> print ("Hello,\nfuture Python programmer!")
```

The output of the code is:

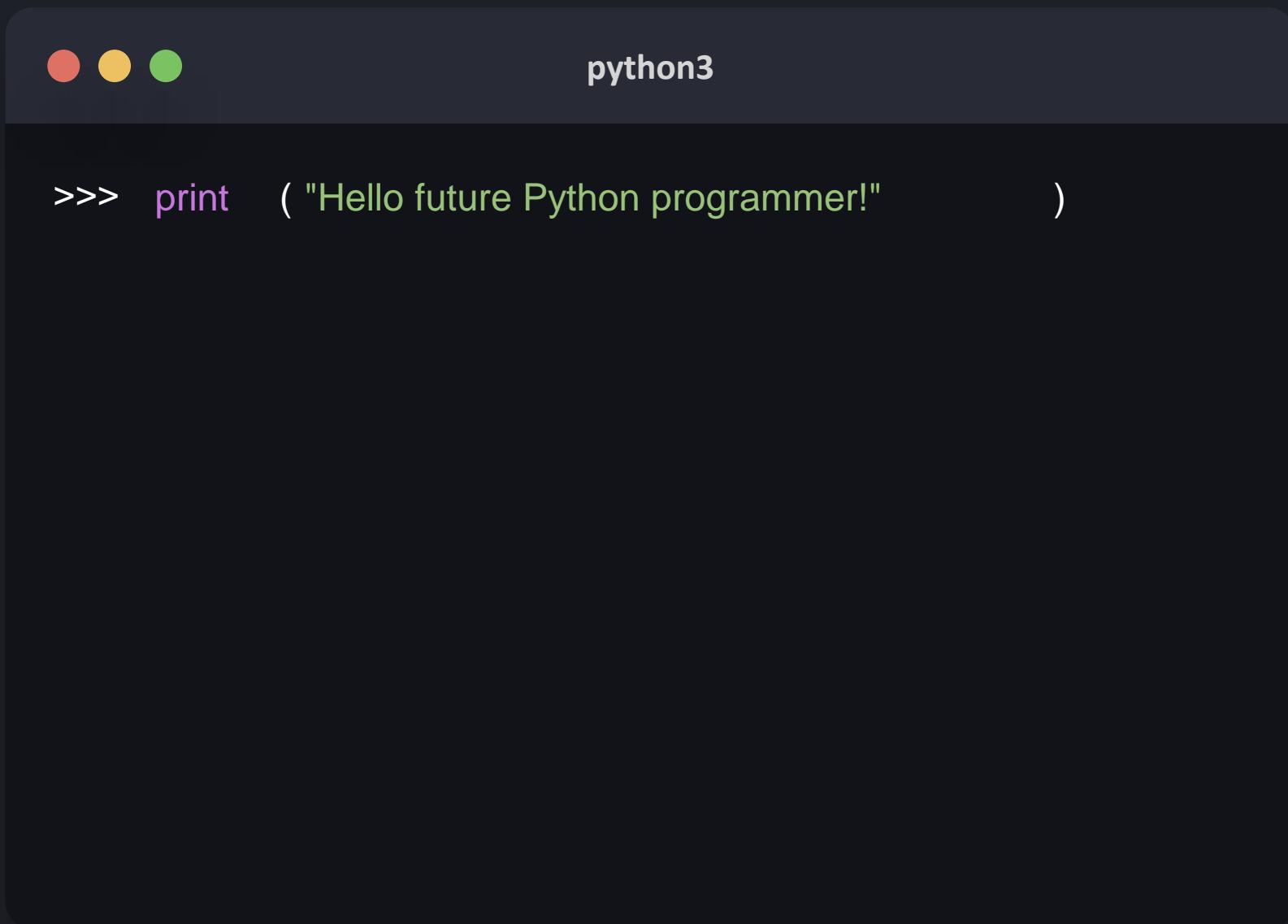
```
Hello,  
future Python programmer!
```

A screenshot of a macOS terminal window. The window title is "python3". In the terminal, the following Python code is run:

```
>>> print ("Hello,\nfuture Python programmer!")
```

The output of the code is:

Hello,
future Python programmer!

A screenshot of a macOS terminal window. The window title is "python3". In the terminal, the following Python code is displayed:

```
>>> print ("Hello future Python programmer!")
```

The text "Hello future Python programmer!" is printed in green.



python3

```
>>> print ( "Hello", "future", "Python", "programmer!" )
```



python3

```
>>> print ( "Hello", "future", "Python", "programmer!" )  
Hello future Python programmer!
```



python3

```
>>> print("Hello!")  
      )  
print("Python is a great language")
```



python3

```
>>> print("Hello!"      , end="")
      print("Python is a great language")
```



python3

```
>>> print("Hello!"      , end="")
      print("Python is a great language")
```



python3

```
>>> print("Hello!"      , end="")
      print("Python is a great language")
Hello!Python is a great language
```



python3

```
>>> print("Hello!"      , end="")
      print("Python is a great language")
Hello!Python is a great language

>>> print("Hello!"      , end="!")
      print("Python is a great language")
Hello!!Python is a great language

>>> print("Hello!"      , end=" ❤ ")
      print("Python is a great language")
Hello! ❤ Python is a great language
```



python3

```
>>> print ( "Hello", "future", "Python", "programmer!" )
```



python3

```
>>> print ( "Hello", "future", "Python", "programmer!" , sep="-" )  
Hello-future-Python-programmer!
```

```
>>> print      "Hello", "future", "Python", "programmer!" , sep="♥" )  
Hello ♥ future ♥ Python ♥ programmer!
```



python3

```
>>> print ( "Hi", "Hello"      , sep="! "      , end=" ❤\n"      )
       print ( "So", "enjoying python?"      , sep= " "      , end=" 😊"  )
```

Hi! Hello ❤

So, enjoying Python? 😊

print()

- Built-in function: can be used without importing it.
- Allows us to print values to the console
- We can invoke it with parentheses.
- We can pass the value we want to print as arguments between the parentheses.
- The backslash \ tells python that the next character has a special meaning (eg. \n)
- Keyword arguments such as **sep** and **end** can be used to format the output.



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Literals

200

"Hello!"

"Python"

-89

200

"Hello!"

"Python"

-89

name

c

age

print

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

200

1298901

-90

1_000_000

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

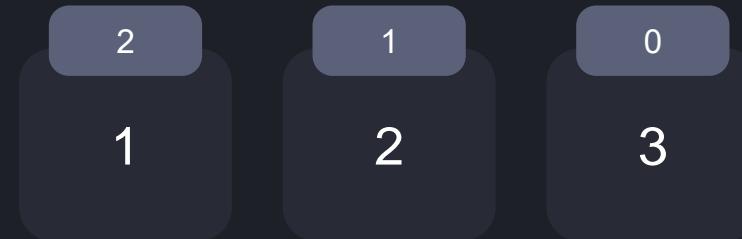
0o123

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

0o123

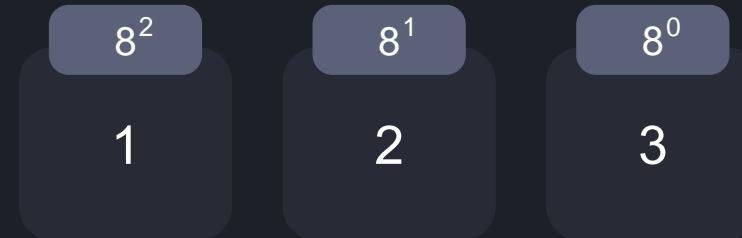


Literals

0o123

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

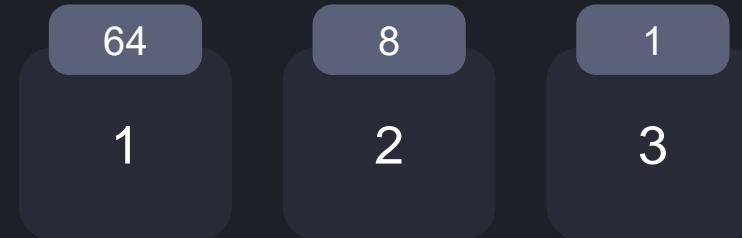


Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

0o123



Literals

0o123

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

64

16

3

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

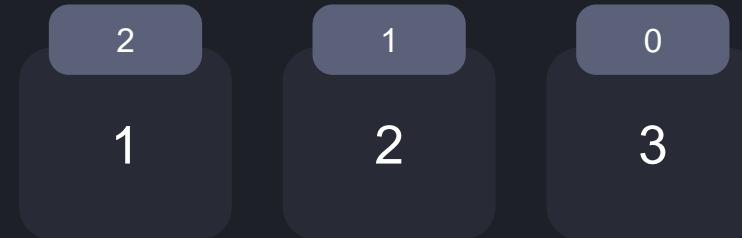
0x123

Literals

0x123

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

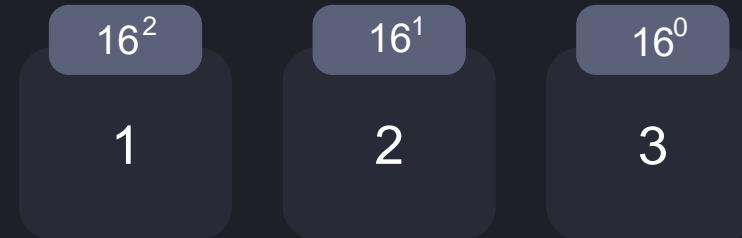


Literals

0x123

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

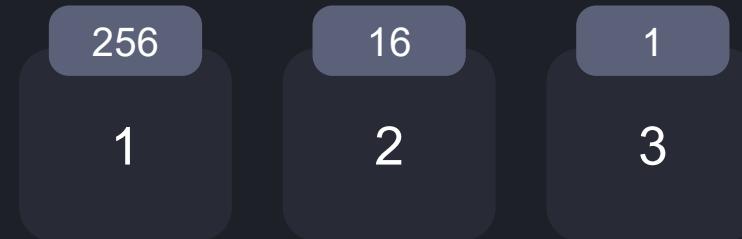


Literals

0x123

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans



Literals

0x123

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

256

32

3

Literals

0x123

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

291

Literals

Literal types:

1. Integers

- Octal numbers

- Hexadecimal numbers

2. Floating point numbers

3. Strings

4. Booleans

45.50

12.1

-90.0

89.394

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
 2. Floating point numbers
 3. Strings
 4. Booleans

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

1e-22

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

"Hello!"

'Hello!'

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

'Hello! "Python" is cool'

"Hello! 'Python' is cool"

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

```
"Hello! \"Python\" is cool"
```

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans

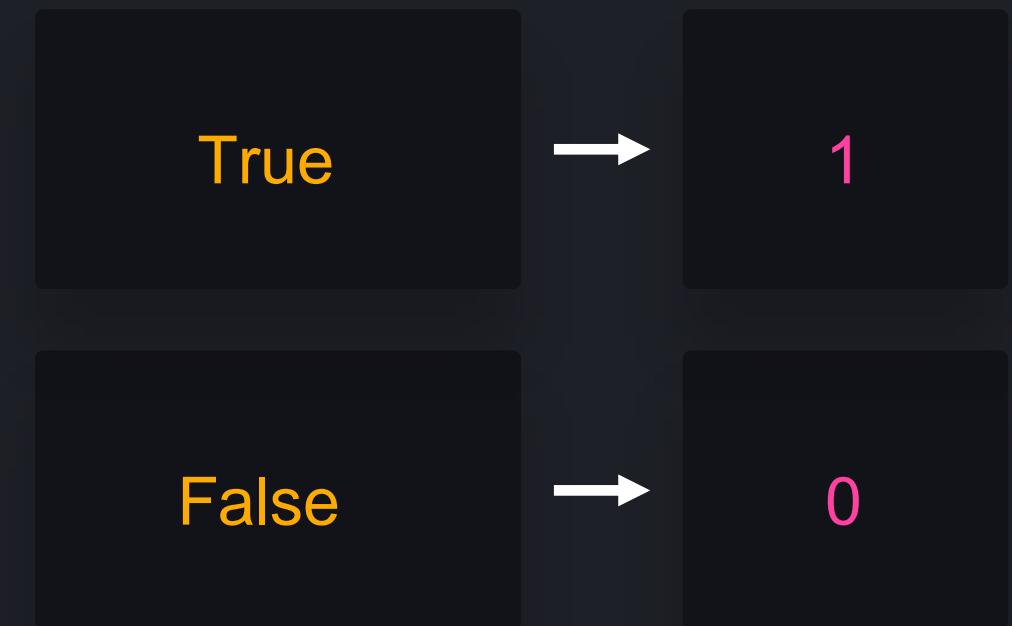
True

False

Literals

Literal types:

1. Integers
 - Octal numbers
 - Hexadecimal numbers
2. Floating point numbers
3. Strings
4. Booleans



Literals

Numbers

- Integers

123

- Octal
|

0o123

- Hexadecimal

0x123

- Floating point

123.45

Strings

- Double quotes

"Hello!"

- Single quotes

'Hello!'

- Use quotes
within strings

'Hi "hi"'

- Escape quotes

'Hi \'hi\'"

Boolean

- False

False

- True

True

- Numeric false

0

- Numeric true

1



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Operators

Arithmetic Operators

+

-

*

/

//

%

**

Add

Subtract

Multiply

Divide

Floor Divide

Modulo

Exponential

Arithmetic Operators

**

Exponential

2³

Arithmetic Operators

**

Exponential

2 ** 3

Arithmetic Operators

**

Exponential



python3

```
>>> print(2 ** 3)
```

```
8
```

```
>>> print(2. ** 3.)
```

```
8.0
```

```
>>> print(2 ** 3.)
```

```
8.0
```

```
>>> print(2. ** 3)
```

```
8.0
```

Arithmetic Operators

*

Multiplication



python3

```
>>> print(2 * 3)
```

```
6
```

```
>>> print(2. * 3.)
```

```
6.0
```

```
>>> print(2 * 3.)
```

```
6.0
```

```
>>> print(2. * 3)
```

```
6.0
```

Arithmetic Operators

/

Division

```
>>> print(10 / 2)
```

```
5.0
```

```
>>> print(10. / 2.)
```

```
5.0
```

```
>>> print(10 / 2.)
```

```
5.0
```

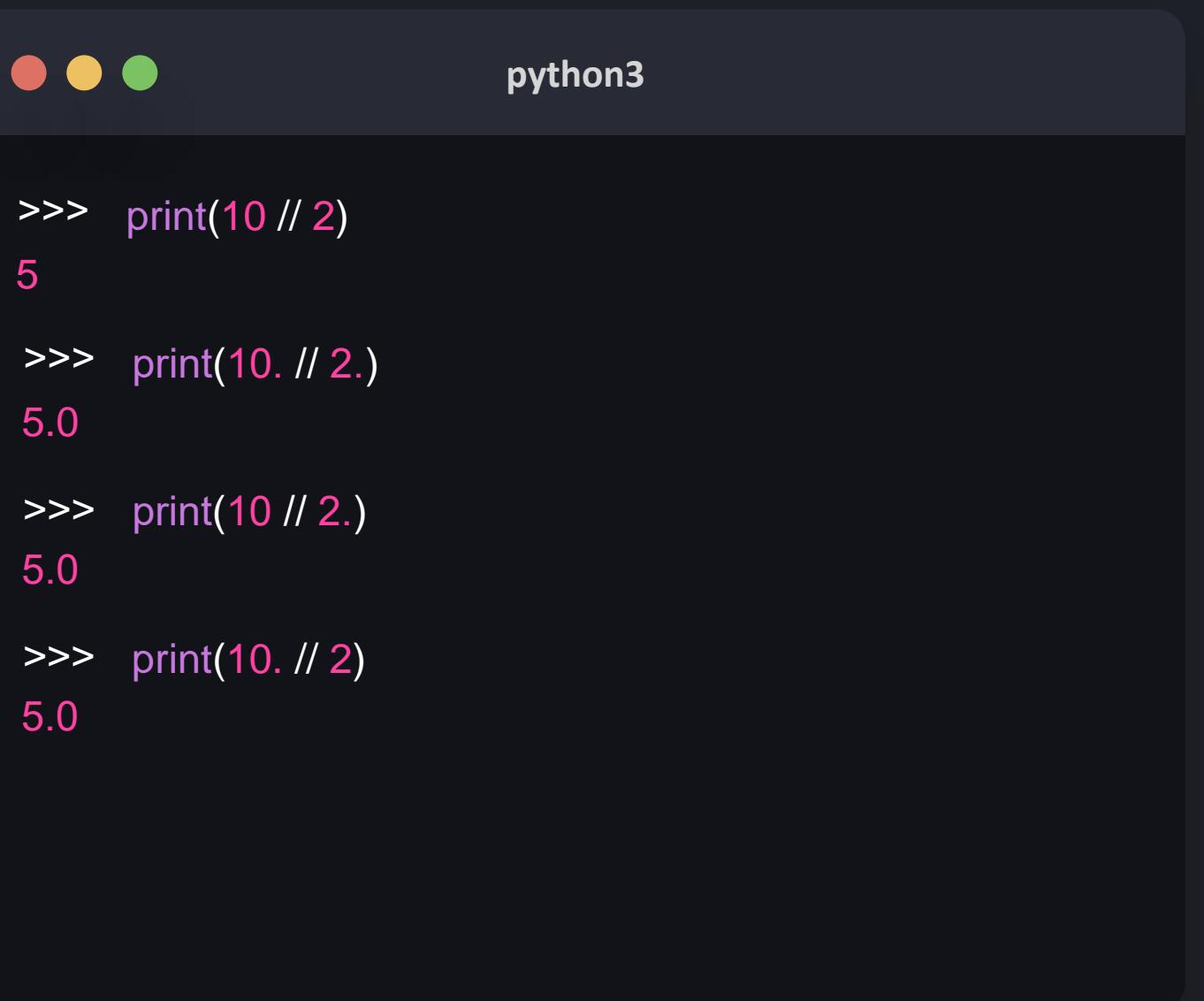
```
>>> print(10. / 2)
```

```
5.0
```

Arithmetic Operators

//

Floor Division



A screenshot of a dark-themed Python terminal window titled "python3". The window shows four examples of the floor division operator (//). The first two examples use integers, resulting in integer outputs (5 and 5.0 respectively). The last two examples use floating-point numbers, also resulting in floating-point outputs (5.0 and 5.0 respectively). The terminal has three colored status indicators at the top: red, yellow, and green.

```
>>> print(10 // 2)
5
>>> print(10. // 2.)
5.0
>>> print(10 // 2.)
5.0
>>> print(10. // 2)
5.0
```

Arithmetic Operators

//

Floor Division



python3

```
>>> print(6. / 4)
```

```
1.5
```

```
>>> print(6. // 4)
```

```
1.0
```

```
>>> print(6. / -4)
```

```
-1.5
```

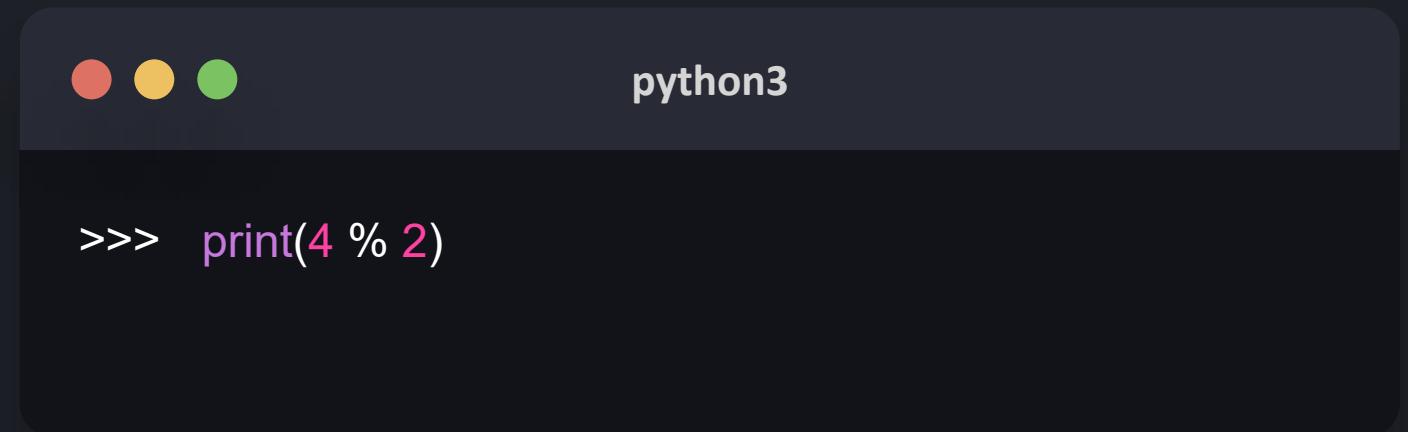
```
>>> print(6. // -4)
```

```
-2.0
```

Arithmetic Operators

%

Modulo



A screenshot of a dark-themed terminal window titled "python3". The window shows the command line prompt ">>> print(4 % 2)" and its output "2". The terminal has three colored window control buttons (red, yellow, green) at the top.

```
>>> print(4 % 2)
```

Arithmetic Operators

%

Modulo



A screenshot of a dark-themed terminal window titled "python3". The window shows the command `>>> print(4 % 2)` and its output `0`. The terminal has three colored window control buttons (red, yellow, green) at the top.

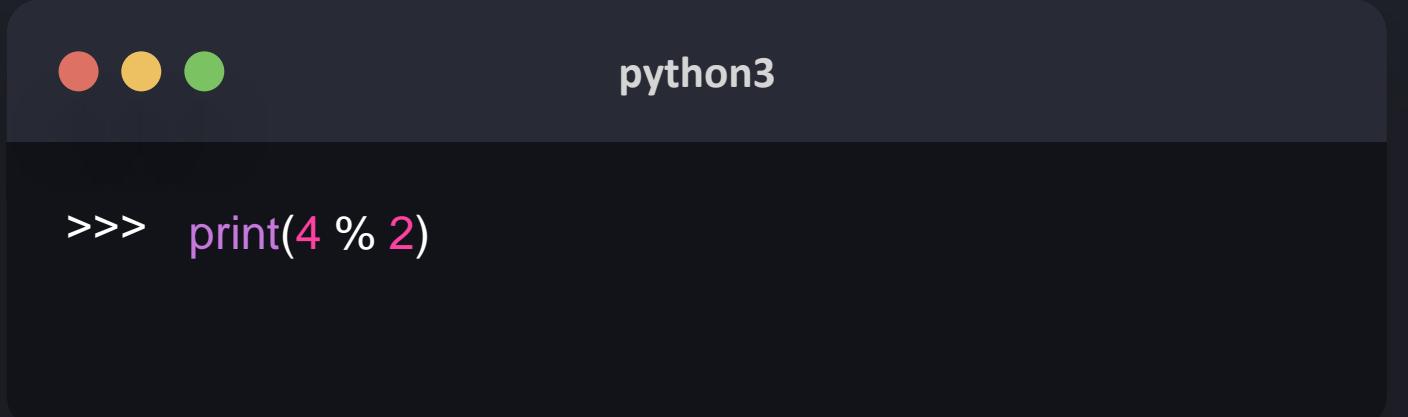
```
>>> print(4 % 2)
```

0

Arithmetic Operators

%

Modulo



A screenshot of a dark-themed terminal window titled "python3". The window shows the Python interpreter prompt ">>>" followed by the code "print(4 % 2)". The output of the code, "0", is displayed below the code. The terminal has three colored window control buttons (red, yellow, green) at the top.

```
>>> print(4 % 2)
```

Arithmetic Operators

%

Modulo



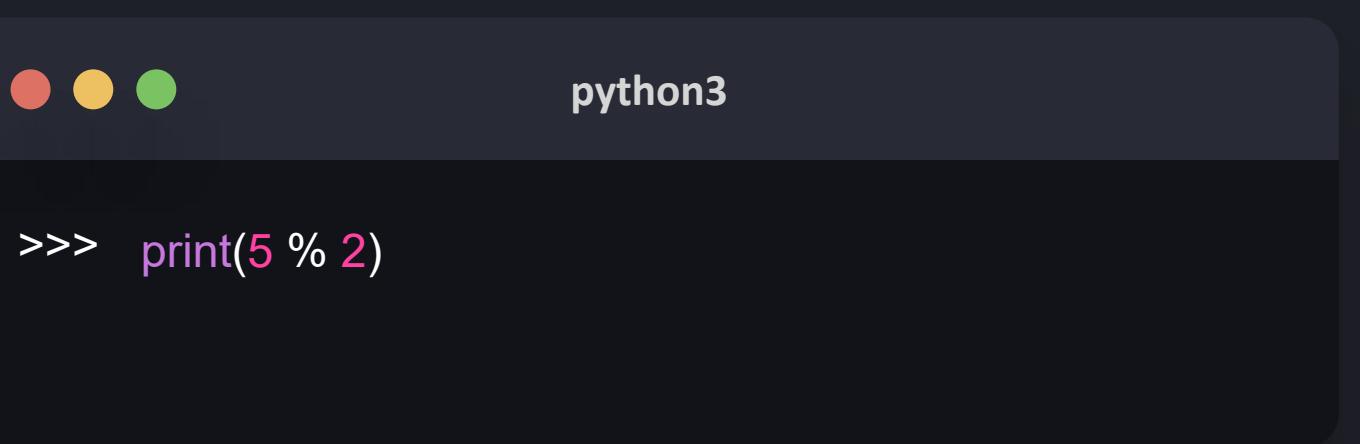
A screenshot of a dark-themed terminal window titled "python3". The window shows the command "print(4 % 2)" being run, and the output "0" is displayed below it. The terminal has three colored window control buttons (red, yellow, green) at the top.

```
>>> print(4 % 2)
0
```

Arithmetic Operators

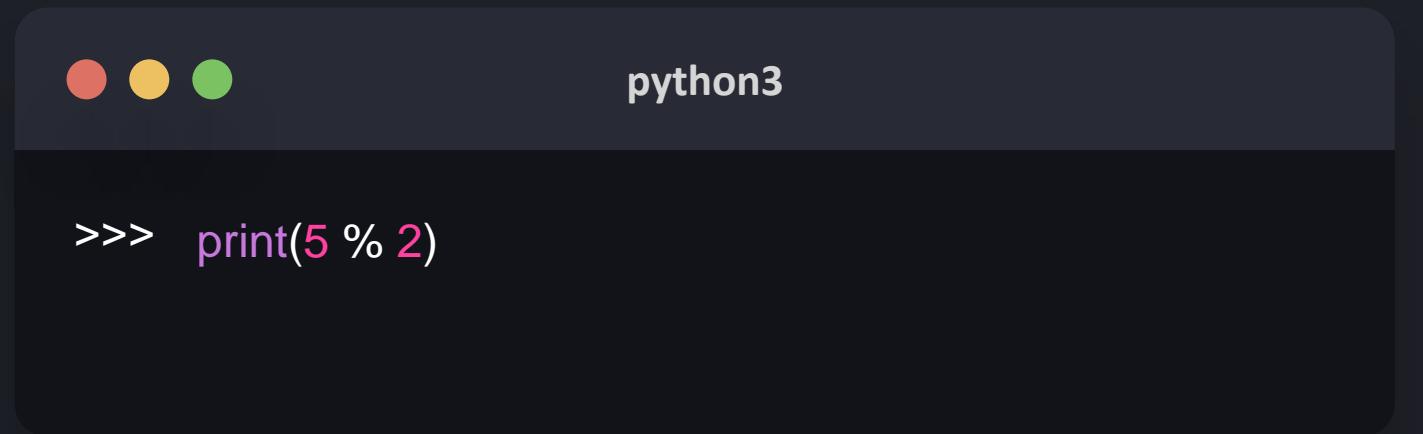
%

Modulo



A screenshot of a dark-themed terminal window titled "python3". The window shows the Python interpreter prompt ">>>" followed by the code "print(5 % 2)". The output of the code, "1", is displayed below the code. The terminal has three colored window control buttons (red, yellow, green) at the top.

```
>>> print(5 % 2)
```

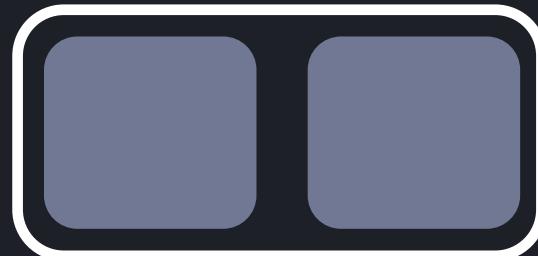


A screenshot of a macOS terminal window titled "python3". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar contains the text "python3". The main area of the terminal shows the Python code: ">>> print(5 % 2)".

Arithmetic Operators

%

Modulo





A screenshot of a macOS terminal window. The window title is "python3". The terminal prompt is "=>>> print(5 % 2)". The output of the command is not visible.

Arithmetic Operators

%

Modulo



Arithmetic Operators

%

Modulo

```
>>> print(5 % 2)  
1
```

python3

Arithmetic Operators

%

Modulo

```
>>> print(5 % 2)  
1
```

python3

Arithmetic Operators

+

Addition



python3

```
>>> print(6 + 4)
```

```
10
```

```
>>> print(6. + 4)
```

```
10.0
```

```
>>> print(6. + 4.)
```

```
10.0
```

Arithmetic Operators

-

Subtraction



python3

```
>>> print(6 - 4)
```

```
2
```

```
>>> print(6. - 4)
```

```
2.0
```

```
>>> print(6. - 4.)
```

```
2.0
```

2

-

2

Binary Operator

2

-

2

Binary Operator

$$2 - 2$$

Binary Operator

2

-

2

Binary Operator

-

2

Unary Operator



python3

```
>>> print(-6 - 6)
```



python3

```
>>> print(-6 - 6)  
-12
```



python3

```
>>> print(-6 - 6)
```

```
-12
```

```
>>> print(10 - -6)
```



python3

```
>>> print(-6 - 6)
```

```
-12
```

```
>>> print(10 - -6)
```

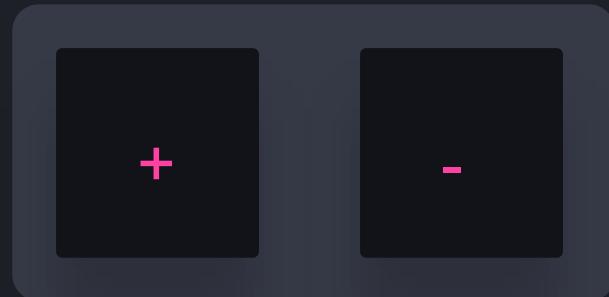
```
16
```



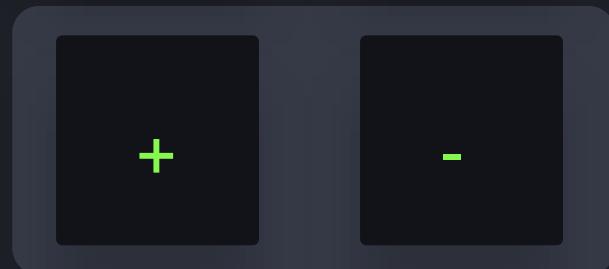
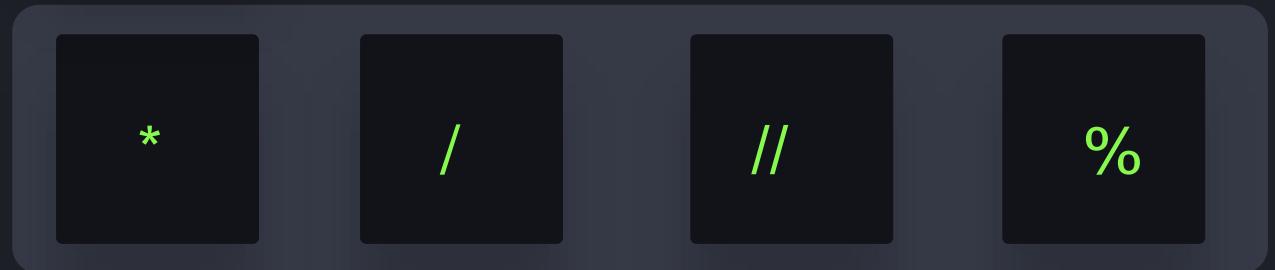
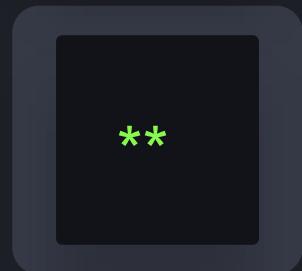
python3

```
>>> print(10 - 6 ** 7 / 9 * 23 + 1)
```

Highest Priority



(unary)



(binary)

Lowest Priority



python3

```
>>> print(10 - 6 ** 2 / 9 * 10 + 1)
```

```
print( 10 - 6 ** 2 / 9 * 10 + 1)
```

```
print( 10 - 6 ** 2 / 9 * 10 + 1)
```

```
print( 10 - 36 / 9 * 10 + 1)
```

```
print( 10 - 4 * 10 + 1)
```

```
print( 10 - 40 + 1)
```

```
print( 10 - 40 + 1)
```

```
print( -30 + 1)
```

```
print( -29 )
```



python3

```
>>> print(10 - 6 ** 7 / 9 * 10 + 1)  
-29
```



python3

```
>>> print(2 *      (2 + 3)    )
```



python3

```
>>> print(2 *      5)
```



python3

```
>>> print(2 *      5)  
10
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Variables



python3

```
>>> print(2 * 5)  
10
```



python3

```
>>> amount_of_apples = 2
>>> cost_of_apple = 5
```



python3

```
>>> amount_of_apples = 2
>>> cost_of_apple = 5
>>> print(amount_of_apples * cost_of_apple)
```



python3

```
>>> amount_of_apples = 2
>>> cost_of_apple = 5
>>> print(amount_of_apples * cost_of_apple)
10
```

amount_of_apples

2

cost_of_apple

5

Valid Variable Names

amount_of_apples
cost_of_apple
_total_cost

Invalid Variable Names

am*unt_o%_apples
c*st_o%_apple
5apples_cost

Valid Variable Names

amount_of_apples
cost_of_apple
_total_cost
COST_OF_APPLE

Invalid Variable Names

am*unt_o%_apples
c*st_o%_apple
5apples_cost

Valid Variable Names

amount_of_apples
cost_of_apple
_total_cost
COST_OF_APPLE

Invalid Variable Names

am*unt_o%_apples
c*st_o%_apple
5apples_cost
del
elif
return

Valid Variable Names

amount_of_apples
cost_of_apple
_total_cost
COST_OF_APPLE

Invalid Variable Names

am*unt_o%_apples
c*st_o%_apple
5apples_cost
del
elif
return

Reserved Keywords

False	elif	lambda
None	else	nonlocal
True	except	not
and	finally	or
as	for	pass
assert	from	raise
break	global	return
class	if	try
continue	import	while
def	in	with
del	is	yield

Valid Variable Names

Import
Del
Elif
Return

Invalid Variable Names

import
del
elif
return

amount_of_apples

2

cost_of_apple

5



python3

```
>>> cost_of_apple = cost_of_apple + 2
```

amount_of_apples

2

cost_of_apple

7



python3

```
>>> cost_of_apple = cost_of_apple + 2
```

amount_of_apples

2

cost_of_apple

7



python3

```
>>> cost_of_apple = cost_of_apple + 2
>>> print(amount_of_apples * cost_of_apple)
```

amount_of_apples

2

cost_of_apple

7



python3

```
>>> cost_of_apple = cost_of_apple + 2
>>> print(amount_of_apples * cost_of_apple)
18
```

amount_of_apples

2

cost_of_apple

5

```
python3  
cost_of_apple = cost_of_apple + 2  
>>>  
>>> print(amount_of_apples * cost_of_apple)  
14
```

amount_of_apples

2

cost_of_apple

5

...
cost_of_apple ^{python3} += 2

>>>

>>> print(amount_of_apples * cost_of_apple)

14

amount_of_apples

2

cost_of_apple

5



python3

```
>>> cost_of_apple += 2
```

amount_of_apples

2

cost_of_apple

7



python3

```
>>> cost_of_apple += 2
>>> print(amount_of_apples * cost_of_apple)
```

amount_of_apples

2

cost_of_apple

7



python3

```
>>> cost_of_apple += 2
>>> print(amount_of_apples * cost_of_apple)
18
```

Without Shortcut Operator

```
cost_of_apple = cost_of_apple + 2  
cost_of_apple = cost_of_apple - 2  
cost_of_apple = cost_of_apple * 2  
cost_of_apple = cost_of_apple ** 2  
cost_of_apple = cost_of_apple / 2  
cost_of_apple = cost_of_apple // 2  
cost_of_apple = cost_of_apple % 2
```

With Shortcut Operator

```
cost_of_apple += 2  
cost_of_apple -= 2  
cost_of_apple *= 2  
cost_of_apple **= 2  
cost_of_apple /= 2  
cost_of_apple //= 2  
cost_of_apple %= 2
```

Variables

- Variables allow you to store values
- A variable has a valid name (letters, digits, underscore, not a reserved keyword)
- Python is dynamically typed: variables can be redeclared
- We can use shortcut operators in order to cleanly redeclare a variable
- We can combine text and variables using the `+` operator in the `print` function



python3

```
>>> print("One apple costs: " + cost_of_apple)  
"One apple costs: 5"
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Comments



python3

```
>>> amount_of_apples = 2          # Amount in basket  
                                # The cost of an apple in USD  
>>> cost_of_apple = 5
```



python3

```
>>> amount_of_apples = 2          # Amount in basket  
                                # The cost of an apple in USD  
# Should always be an integer  
>>> cost_of_apple = 5
```



python3

```
>>> amount_of_apples = 2          # Amount in basket  
                                # The cost of an apple in USD  
# Should always be an integer  
>>> cost_of_apple = 5
```



python3

```
>>> amount_of_apples = 2
>>> cost_of_apple = 5
```



python3

```
>>> amount_of_apples = 2
>>> # cost_of_apple = 5
>>> print(amount_of_apples * cost_of_apple)
```



python3

```
>>> amount_of_apples = 2
>>> # cost_of_apple = 5
>>> print(amount_of_apples * cost_of_apple)
NameError: name 'cost_of_apple' is not defined
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Input



python3

```
>>> print("Hello!")  
Hello!
```

input()



python3

```
>>> input("How are you feeling today? ")
```



python3

```
>>> input("How are you feeling today? ")  
How are you feeling today?           Fantastic!
```



python3

```
>>> input("How are you feeling today? ")  
How are you feeling today?           Fantastic!  
>>>
```



python3

```
>>> favorite_color = input("What is your favorite color? ")
```



python3

```
>>> favorite_color = input("What is your favorite color? ")
```

```
What is your favorite color? blue
```



python3

```
>>> favorite_color = input("What is your favorite color? ")
```

```
What is your favorite color? blue
```

```
>>> print("Your favorite color is " + favorite_color)
```



python3

```
>>> favorite_color = input("What is your favorite color? ")
```

```
What is your favorite color? blue
```

```
>>> print("Your favorite color is " + favorite_color)
```

```
Your favorite color is blue
```



python3

```
>>> age = input("How old are you? ")  
How old are you? 22
```



python3

```
>>> age = input("How old are you? ")  
How old are you?      22  
>>> print(age - 10)  
TypeError: unsupported operand type(s) for -: 'str' and 'int'
```



python3

```
>>> age = input("How old are you? ")  
How old are you? 22
```

Type Casting

- Integers

`int()`

- Floating point

`float()`



`python3`

```
>>> age = input("How old are you? ")
```

```
How old are you?      22
```

```
>>> print(int(age) - 10)
```

Type Casting

- Integers

`int()`

- Floating point

`float()`



`python3`

```
>>> age = input("How old are you? ")
```

```
How old are you?      22
```

```
>>> print(int(age) - 10)
```

```
12
```

Type Casting

- Integers

`int()`

- Floating point

`float()`



`python3`

```
>>> age = int(input("How old are you? "))
```

```
How old are you?      22
```

```
>>> print(age - 10)
```

```
12
```

input()

- Prompts the user to input some data from the console
- It accepts an optional parameter that can be used in order to write a message before the user input
- Always returns a string
- A program that doesn't use any input function, is called a **deaf program**

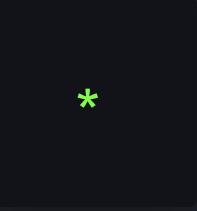


KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



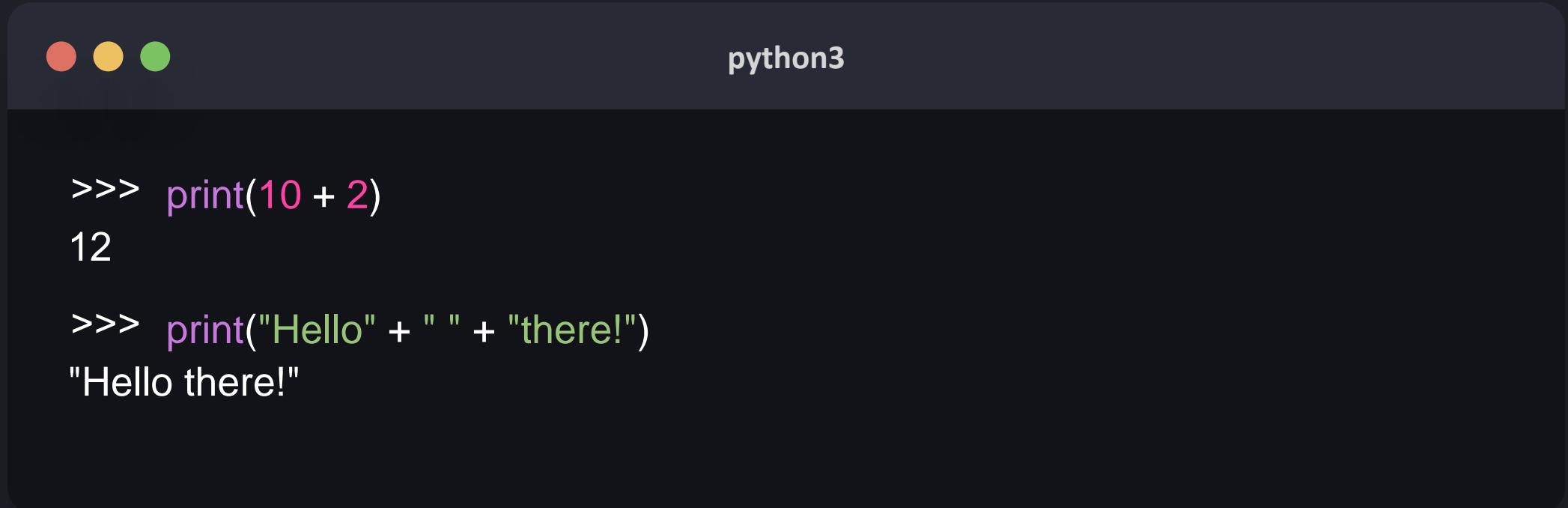
String Methods

 + *

python3

```
>>> print(10 + 2)  
12
```

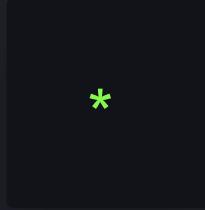
```
>>> print(10 * 2)  
22
```



A dark-themed terminal window titled "python3". The window has three colored circular icons in the top-left corner: red, yellow, and green. The title bar also contains the text "python3". The main area of the terminal displays the following Python code and its output:

```
>>> print(10 + 2)
12

>>> print("Hello" + " " + "there!")
"Hello there!"
```



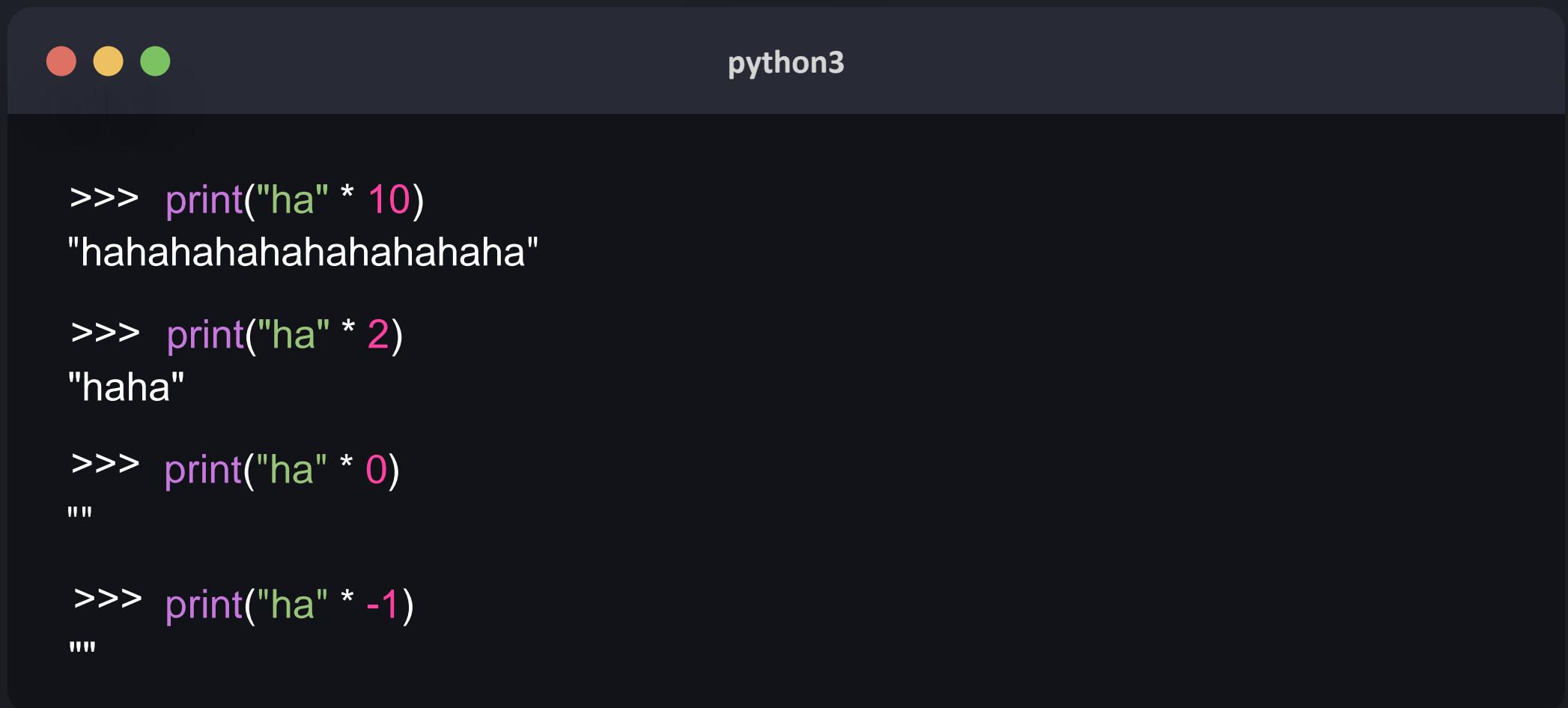
```
python3
>>> print(10 * 2)
22
```



python3

```
>>> print(10 * 2)
22

>>> print("ha" * 10)
"hahahahahahahahaha"
```



```
*  
python3  
  
>>> print("ha" * 10)  
"hahahahahahahahaha"  
  
>>> print("ha" * 2)  
"haha"  
  
>>> print("ha" * 0)  
"  
  
>>> print("ha" * -1)  
""
```



python3

```
>>> print(int("22"))
22
```



python3

```
>>> print(int("22"))
```

```
22
```

```
>>> print(str(22))
```

```
"22"
```



python3

```
>>> cost_of_apple = 2
>>> amount_of_apples = input("How many apples do you want? ")
How many apples do you want? 10
```



python3

```
>>> cost_of_apple = 2
>>> amount_of_apples = input("How many apples do you want? ")
How many apples do you want? 10
```



python3

```
>>> cost_of_apple = 2
>>> amount_of_apples = input("How many apples do you want? ")
How many apples do you want? 10
>>> total_sum = cost_of_apple * int(amount_of_apples)
```



python3

```
>>> cost_of_apple = 2
>>> amount_of_apples = input("How many apples do you want? ")
How many apples do you want? 10
>>> total_sum = cost_of_apple * int(amount_of_apples)
>>> print("You have to pay: " + str(total_sum))
```



python3

```
>>> cost_of_apple = 2
>>> amount_of_apples = input("How many apples do you want? ")
How many apples do you want? 10
>>> total_sum = cost_of_apple * int(amount_of_apples)
>>> print("You have to pay: " + str(total_sum))
You have to pay: 20
```

String Operations

- You can use `+` in order to concatenate two strings
- You can use `*` in order to repeat a string a several amount of times.
- With the `str` function, you can type-cast a number into a string



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Comparison Operators

Comparison Operators

`==`

`!=`

`>`

`>=`

`<`

`<=`

Comparison Operators

`==`

Equal



python3

```
>>> print(2 == 2)
```

True

```
>>> print(2 == 4)
```

False

```
>>> print("Hello!" == "Hello!")
```

True

```
>>> print("Hello!" == "Goodbye!")
```

False

```
>>> print(4 == (2 * 2))
```

True

Comparison Operators

!=

Not Equal



python3

```
>>> print(2 != 2)
```

False

```
>>> print(2 != 4)
```

True

```
>>> print("Hello!" != "Hello!")
```

False

```
>>> print("Hello!" != "Goodbye!")
```

True

```
>>> print(4 != (2 * 2))
```

False

Comparison Operators

>

Greater than

```
python3  
  
>>> print(4 > 2)  
True  
  
>>> print(2 > 4)  
False  
  
>>> print(2 > 2)  
False  
  
>>> cost_of_apple = 2  
>>> cost_of_banana = 3  
  
>>> print(cost_of_apple > cost_of_banana)  
False
```

Comparison Operators

`>=`

Greater than or equal to



python3

```
>>> print(4 >= 2)
```

True

```
>>> print(2 >= 4)
```

False

```
>>> print(2 >= 2)
```

True

Comparison Operators

<

Smaller than



python3

```
>>> print(4 < 2)
```

```
False
```

```
>>> print(2 < 4)
```

```
True
```

```
>>> print(2 < 2)
```

```
False
```

```
>>> cost_of_apple = 2
```

```
>>> cost_of_banana = 3
```

```
>>> print(cost_of_apple < cost_of_banana)
```

```
True
```

Comparison Operators

`<=`

Smaller than or equal to



python3

```
>>> print(4 <= 2)
```

False

```
>>> print(2 <= 4)
```

True

```
>>> print(2 <= 2)
```

True

Comparison Operators

`==`

`!=`

`>`

`>=`

`<`

`<=`



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Conditional Statements

if *condition*:

```
if condition :  
    print("The condition is true!")
```

If **True** :

```
print("The condition is true!")
```

```
if False :  
    print("The condition is true!")
```



python3

```
>>> age = int(input("How old are you? "))
```



python3

```
>>> age = int(input("How old are you? "))
How old are you? 22
```



python3

```
>>> age = int(input("How old are you? "))
```

```
How old are you? 22
```

```
>>> if age >= 18:
```

```
    print("You are an adult!")
```



python3

```
>>> age = int(input("How old are you? "))
```

```
How old are you? 22
```

```
>>> if age >= 18:
```

```
    print("You are an adult!")
```

```
You are an adult!
```

```
if condition :  
    print("The condition is true!")
```

```
if condition:  
    print("The condition is true!")
```

```
if    condition :  
    print("The condition is true!")  
  
else:  
    print("The condition is false!")
```

```
if True :  
    print("The condition is true!")  
  
else:  
    print("The condition is false!")
```

```
if False :  
    print("The condition is true!")
```

```
else:
```

```
print("The condition is false!")
```

```
if    condition:  
    print("The condition is true!")  
  
elif   second_condition :  
    print("Only the second condition is true!")  
  
else:  
    print("Both conditions are false!")
```

```
if False :  
    print("The condition is true!")  
elif True :  
    print("Only the second condition is true!")  
  
else:  
    print("Both conditions are false!")
```

```
if True :  
    print("The condition is true!")  
elif True :  
    print("Only the second condition is true!")  
  
else:  
    print("Both conditions are false!")
```

```
if    age >= 18:  
    if    age == 18:  
        print("You are exactly 18 years old!")  
    else:  
        print("You older than 18 years old!")
```

```
if    age >= 18:  
    if    age == 18:  
        print("You are exactly 18 years old!")  
    else:  
        print("You older than 18 years old!")
```

```
if    age >= 18:  
    if    age == 18:  
        print("You are exactly 18 years old!")  
    else:  
        print("You older than 18 years old!")
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Loops - While

while condition:

while condition:



python3

```
>>> secret_number = 3
>>> guess = int(input("Guess a number: "))
>>> while guess != secret_number:
    guess = int(input("Guess a number: "))
```



python3

```
>>> secret_number = 3
>>> guess = int(input("Guess a number: "))
>>> while guess != secret_number:
    guess = int(input("Guess a number: "))
Guess a number: 0
```



python3

```
>>> secret_number = 3
>>> guess = int(input("Guess a number: "))
>>> while guess != secret_number:
    guess = int(input("Guess a number: "))

Guess a number:    0
Guess a number:    4
```



python3

```
>>> secret_number = 3
>>> guess = int(input("Guess a number: "))
>>> while guess != secret_number:
    guess = int(input("Guess a number: "))

Guess a number: 0
Guess a number: 4
Guess a number: 3
```



python3

```
>>> secret_number = 3
>>> guess = int(input("Guess a number: "))
>>> while guess != secret_number:
    guess = int(input("Guess a number: "))

Guess a number: 0
Guess a number: 4
Guess a number: 3
>>>
```



python3

```
>>> secret_number = 3
>>> guess = int(input("Guess a number: "))
>>> while guess != secret_number:
    guess = int(input("Guess a number: "))
```



python3

```
>>> secret_number = 3
>>> guess = int(input("Guess a number: "))
>>> while guess != secret_number:
    guess = int(input("Guess a number: "))
else:
    print("Congratulations, you got it!")
```



python3

```
>>> secret_number = 3
>>> guess = int(input("Guess a number: "))
>>> while guess != secret_number:
    guess = int(input("Guess a number: "))
else:
    print("Congratulations, you got it!")
```

Guess a number: 0



python3

```
>>> secret_number = 3
>>> guess = int(input("Guess a number: "))
>>> while guess != secret_number:
    guess = int(input("Guess a number: "))
else:
    print("Congratulations, you got it!")
```

Guess a number: 0

Guess a number: 4



python3

```
>>> secret_number = 3
>>> guess = int(input("Guess a number: "))
>>> while guess != secret_number:
    guess = int(input("Guess a number: "))
else:
    print("Congratulations, you got it!")

Guess a number: 0
Guess a number: 4
Guess a number: 3
```



python3

```
>>> secret_number = 3
>>> guess = int(input("Guess a number: "))
>>> while guess != secret_number:
    guess = int(input("Guess a number: "))
else:
    print("Congratulations, you got it!")
```

Guess a number: 0

Guess a number: 4

Guess a number: 3

Congratulations, you got it!

>>>



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Loops - For

for ... in ...:

```
for i in range(9):
```

```
for i in range(9):
```

0

1

2

3

4

5

6

7

8

9

```
for i in range(9):
```

0

1

2

3

4

5

6

7

8

9

i

0

1

2

3

4

5

6

7

8

9



python3

```
>>> for i in range(9):  
    print("i is: ", i)
```

i is: 0

i

0

1

2

3

4

5

6

7

8

9



python3

```
>>> for i in range(9):
    print("i is: ", i)

i is: 0
i is: 1
```

0 1 2 3 4 5 6 7 8 9



python3

```
>>> for i in range(9):  
    print("i is: ", i)
```

```
i is: 0  
i is: 1  
i is: 2
```

0 1 2 3 4 5 6 7 8 9

i



python3

```
>>> for i in range(9):
    print("i is: ", i)

i is: 0
i is: 1
i is: 2
i is: 3
```

0

1

2

3

4

5

6

7

8

9

i



python3

```
>>> for i in range(9):  
    print("i is: ", i)
```

i is: 0

i is: 1

i is: 2

i is: 3

i is: 4

0 1 2 3 4 5 6 7 8 9

i



python3

```
>>> for i in range(9):  
    print("i is: ", i)
```

i is: 0

i is: 1

i is: 2

i is: 3

i is: 4

i is: 5

0 1 2 3 4 5 6 7 8 9

i



python3

```
>>> for i in range(9):  
    print("i is: ", i)
```

i is: 0

i is: 1

i is: 2

i is: 3

i is: 4

i is: 5

i is: 6

0 1 2 3 4 5 6 7 8 9

i



python3

```
>>> for i in range(9):  
    print("i is: ", i)
```

i is: 0

i is: 1

i is: 2

i is: 3

i is: 4

i is: 5

i is: 6

i is: 7

i

0 1 2 3 4 5 6 7 8 9



python3

```
>>> for i in range(9):  
    print("i is: ", i)
```

i is: 0

i is: 1

i is: 2

i is: 3

i is: 4

i is: 5

i is: 6

i is: 7

i is: 8

i

0

1

2

3

4

5

6

7

8

9



python3

```
>>> for i in range(9):  
    print("i is: ", i)
```

```
i is: 0
```

```
i is: 1
```

```
i is: 2
```

```
i is: 3
```

```
i is: 4
```

```
i is: 5
```

```
i is: 6
```

```
i is: 7
```

```
i is: 8
```

```
i is: 9
```

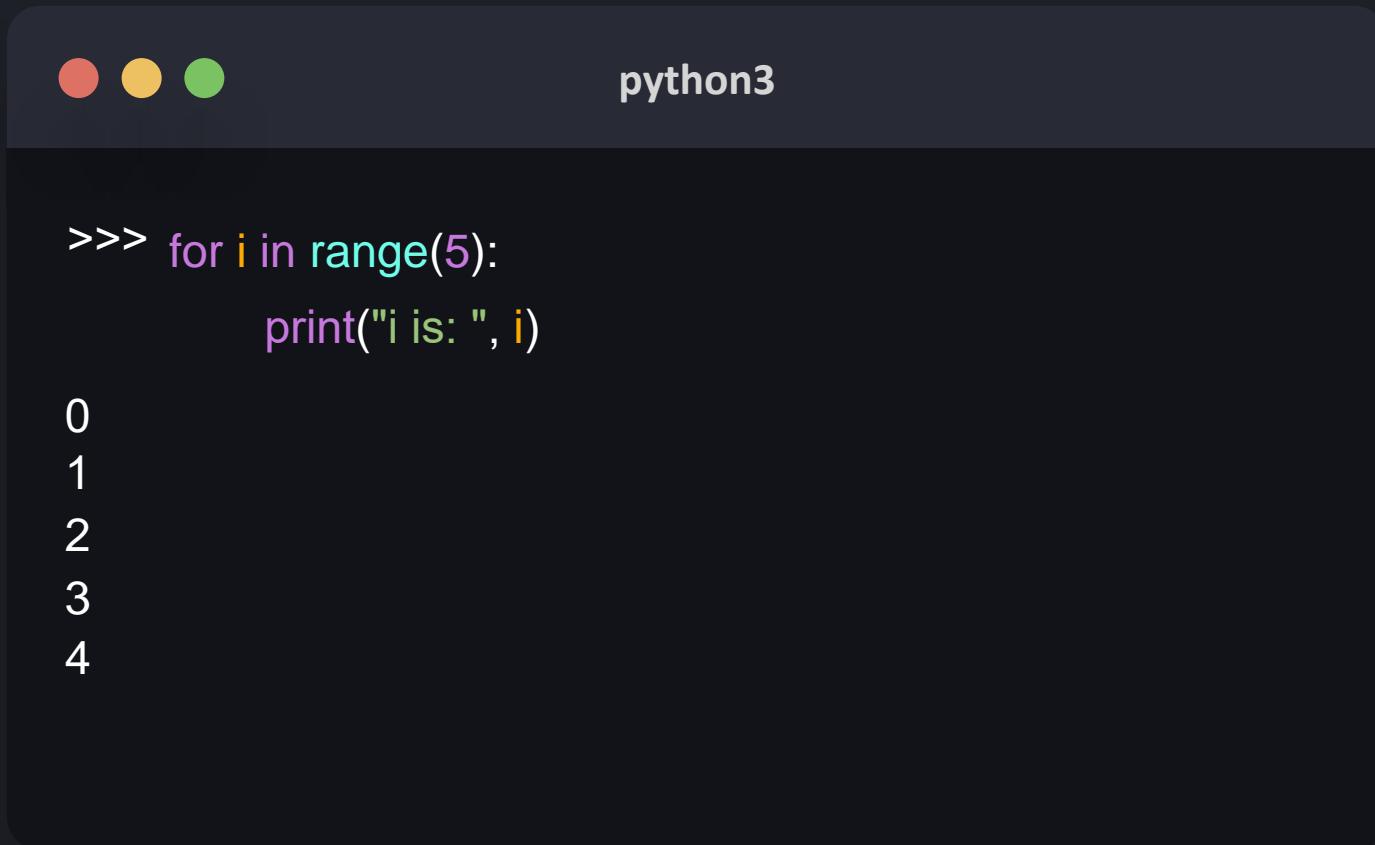
```
for i in range(2, 5):
```

2

3

4

0 1 2 3 4



A screenshot of a macOS terminal window. The window title is "python3". The terminal prompt is ">>>". Below it, a for loop is executed:

```
>>> for i in range(5):
    print("i is: ", i)
```

The output of the loop is:

```
0
1
2
3
4
```

0

1

2

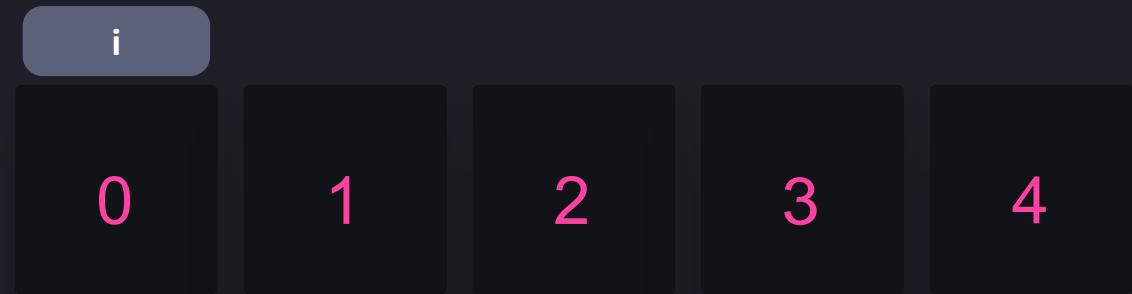
3

4



python3

```
>>> for i in range(5):
    if(i == 2):
        break
    print("i is: ", i)
```

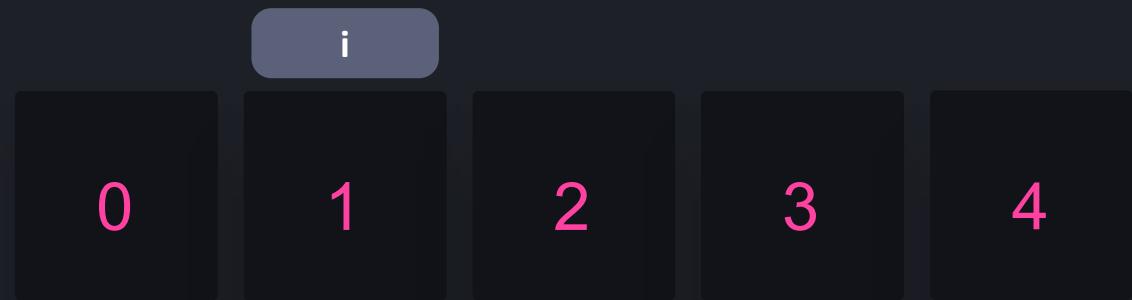


A screenshot of a terminal window titled "python3". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar also displays "python3". The terminal window contains the following Python code:

```
>>> for i in range(5):
    if(i == 2):
        break
    print("i is: ", i)
```

The output of the code is:

```
0
```

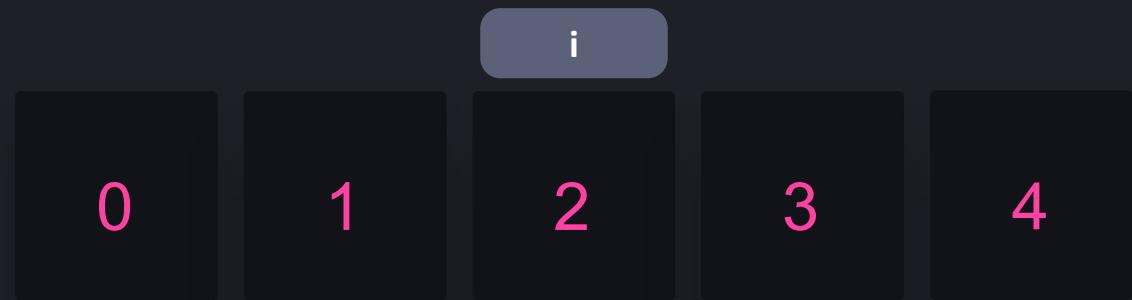


A screenshot of a terminal window titled "python3". The window has three colored window control buttons (red, yellow, green) at the top left. The terminal displays the following Python code:

```
>>> for i in range(5):
    if(i == 2):
        break
    print("i is: ", i)
```

The output of the code is:

```
0
1
```



A screenshot of a macOS terminal window titled "python3". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar shows "python3". The terminal window contains the following Python code:

```
>>> for i in range(5):
    if(i == 2):
        break
    print("i is: ", i)
```

The output of the code is:

```
0
1
>>>
```

0

1

2

3

4



```
>>> for i in range(5):
    print("i is: ", i)
```

0

1

2

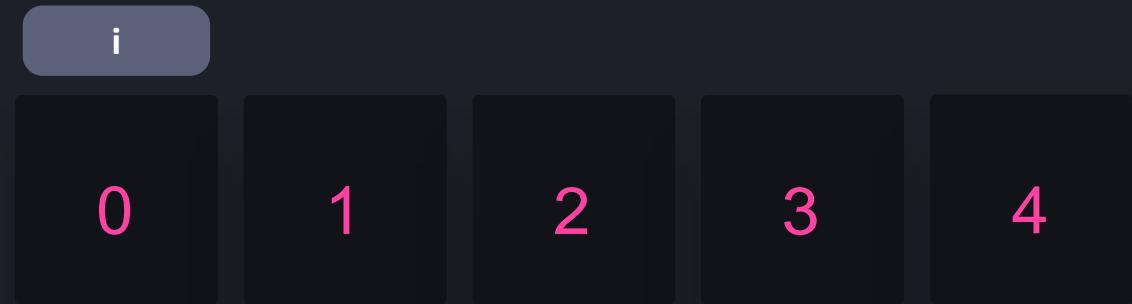
3

4



python3

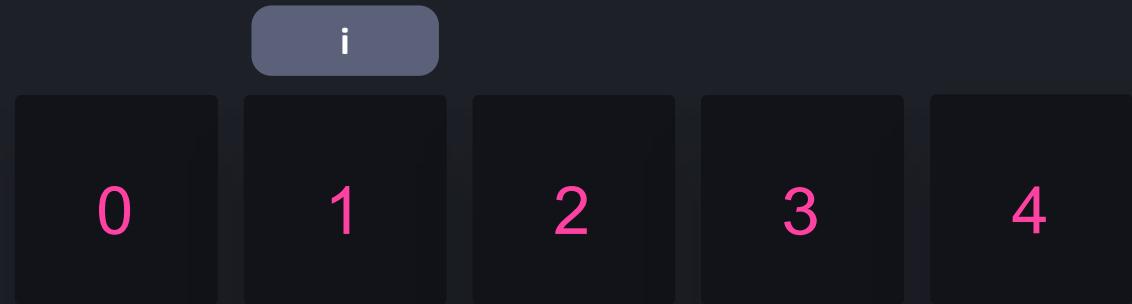
```
>>> for i in range(5):
    if(i == 2):
        continue
    print("i is: ", i)
```



A screenshot of a terminal window titled "python3". The window has three colored window control buttons (red, yellow, green) on the left. The terminal displays the following Python code:

```
>>> for i in range(5):
    if(i == 2):
        continue
    print("i is: ", i)
```

The output of the code is "0", indicating that the value of 'i' is 0 when the loop begins.

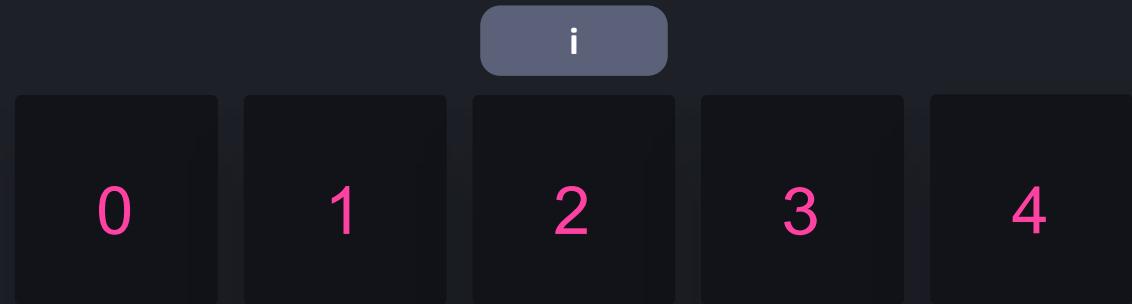


A screenshot of a terminal window titled "python3". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar contains the text "python3". The terminal window displays the following Python code:

```
>>> for i in range(5):
    if(i == 2):
        continue
    print("i is: ", i)
```

The output of the code is:

```
0
1
```



A screenshot of a terminal window titled "python3". The window has three colored window control buttons (red, yellow, green) at the top left. The terminal displays the following Python code:

```
>>> for i in range(5):
    if(i == 2):
        continue
    print("i is: ", i)
```

The output of the code is:

```
0
1
```

i

0

1

2

3

4



python3

```
>>> for i in range(5):
    if(i == 2):
        continue
    print("i is: ", i)
```

0

1

3

i

0

1

2

3

4



python3

```
>>> for i in range(5):
    if(i == 2):
        continue
```

```
    print("i is: ", i)
```

0
1
3
4

- `if/else` statements allow us to conditionally run code
- A `while` loop makes it possible to repetitively execute code based on a certain condition
- We can execute code for each item in a sequence with a `for ... in` loop



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Operators



python3

```
>>> age1 = 24
```



python3

```
>>> age1 = 24  
>>> age2 = 16
```



python3

```
>>> age1 = 24
>>> age2 = 16
>>> if(      Both ages are higher than 18      ):
    print("You are both adults")
    elif(      One age is higher than 18      ):
        print("One of you is an adult")
    else:
        print("You are both children")
```



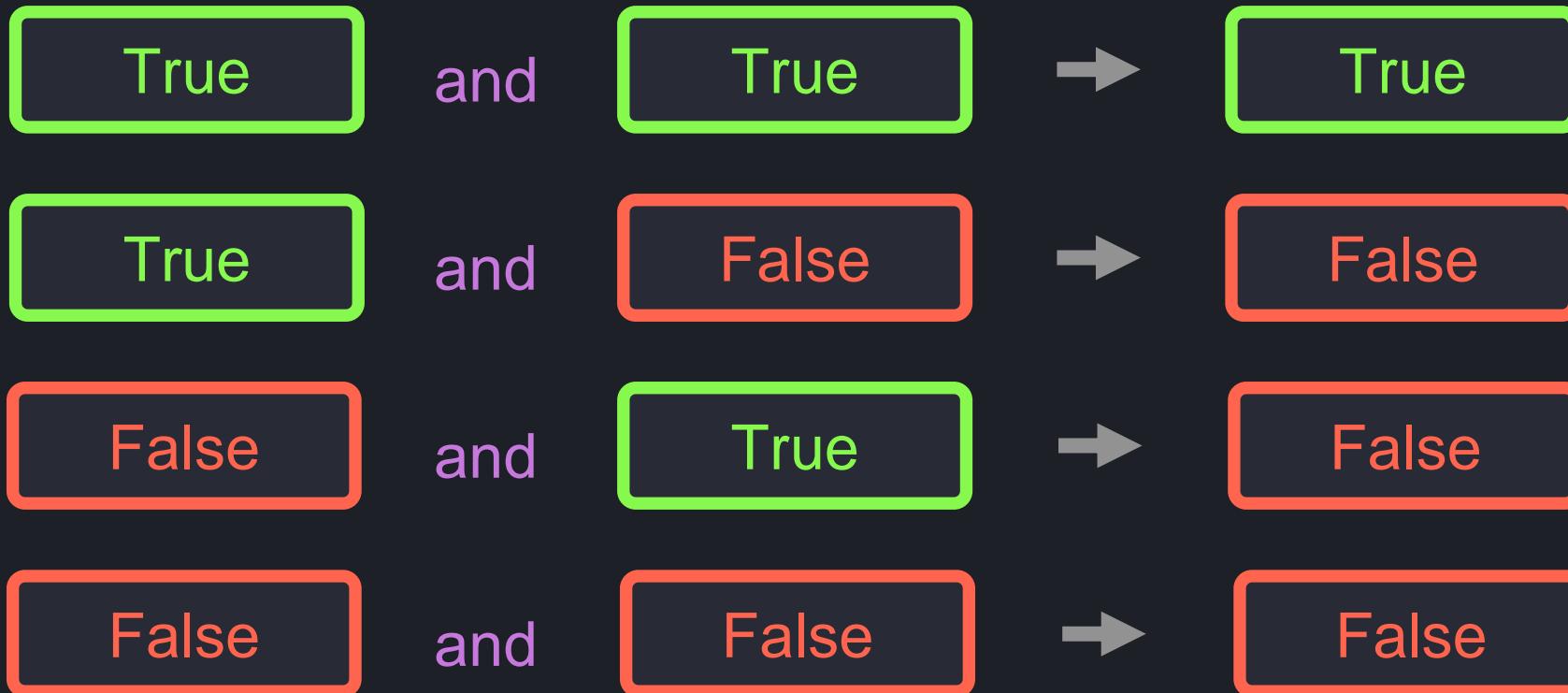
python3

```
>>> age1 = 24
>>> age2 = 16
>>> if(      Both ages are higher than 18      ):
    print("You are both adults")
    elif(      One age is higher than 18      ):
        print("One of you is an adult")
    else:
        print("You are both children")
```



python3

```
>>> age1 = 24
>>> age2 = 16
>>> if(  age1 >= 18    and  age2 >= 18  ):
    print("You are both adults")
    elif(      One age is higher than 18      ):
        print("One of you is an adult")
    else:
        print("You are both children")
```





python3

```
>>> age1 = 24
>>> age2 = 16
>>> if(  age1 >= 18    and   age2 >= 18  ):
    print("You are both adults")
    elif(      One age is higher than 18      ):
        print("One of you is an adult")
    else:
        print("You are both children")
```



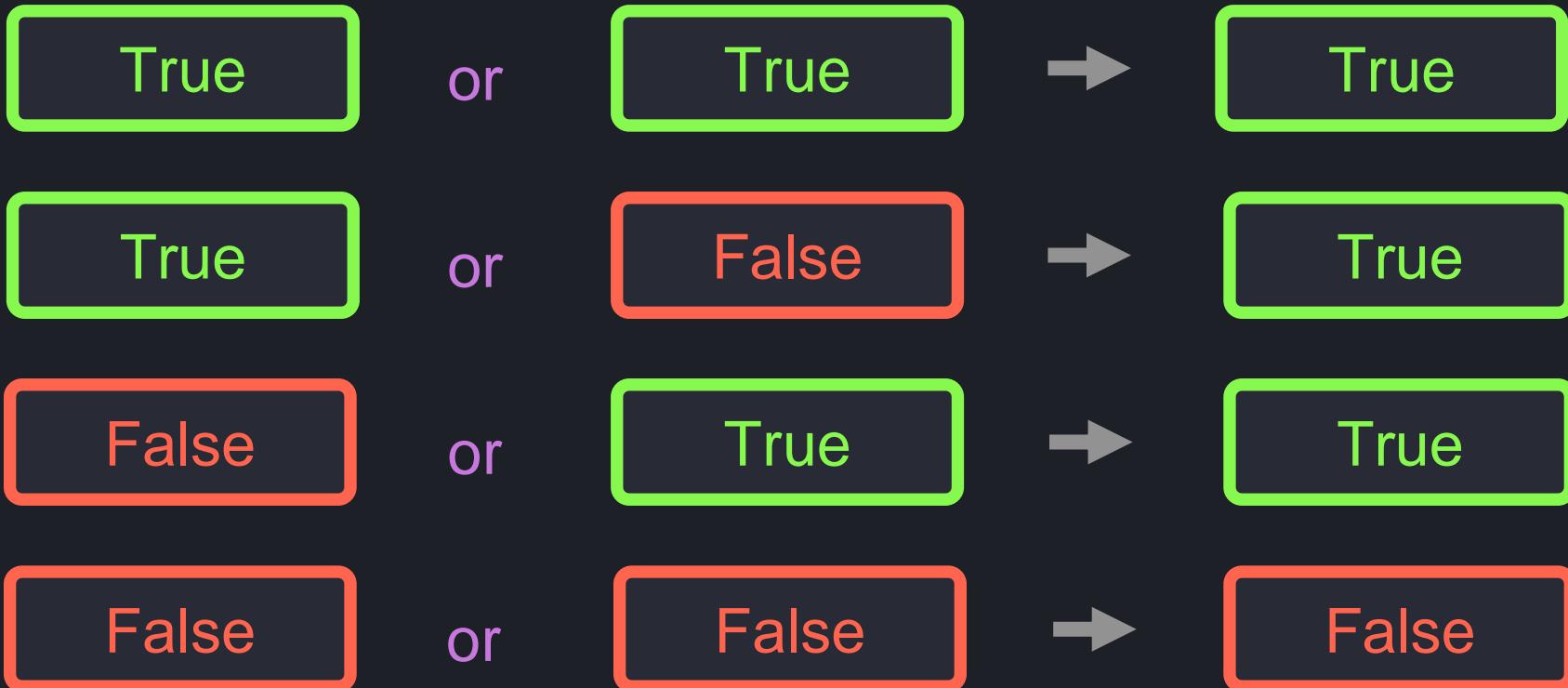
python3

```
>>> age1 = 24
>>> age2 = 16
>>> if(  age1 >= 18    and   age2 >= 18  ):
    print("You are both adults")
    elif(      One age is higher than 18      ):
        print("One of you is an adult")
    else:
        print("You are both children")
```



python3

```
>>> age1 = 24
>>> age2 = 16
>>> if(    age1 >= 18      and    age2 >= 18    ):
        print("You are both adults")
    elif(    age1 >= 18      or       age2 >= 18    ):
        print("One of you is an adult")
    else:
        print("You are both children")
```





python3

```
>>> age1 = 24
>>> age2 = 16
>>> if(    age1 >= 18      and    age2 >= 18    ):
    print("You are both adults")
elif(    age1 >= 18      or       age2 >= 18    ):
    print("One of you is an adult")
else:
    print("You are both children")
```





python3

```
>>> is_hungry = False
>>> if( not is_hungry ):
    print("You are not hungry")
```



python3

```
>>> is_hungry = False
>>> if( not is_hungry ):
    print("You are not hungry")
"You are not hungry"
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Bitwise Operators

Logical Operators

or

and

not

Bitwise Operators

&

|

~

^

Conjunction

Disjunction

Negation

Exclusive

Bitwise Operators

&

Two 1

$$\boxed{1} \text{ & } \boxed{1}$$

|

At least one 1

$$\begin{array}{c|c} \boxed{1} & | \\ \hline \boxed{0} & | \\ \hline \boxed{1} & | \\ \hline \end{array}$$

^

Exactly 1

$$\begin{array}{c|c} \boxed{0} & ^ \\ \hline \boxed{1} & ^ \\ \hline \end{array}$$



python3

```
>>> 15 & 22
```



python3

```
>>> print(bin(15))
0b1111
```



python3

```
>>> print(bin(15))
```

```
0b1111
```

```
>>> print(bin(22))
```

```
0b10110
```



python3

```
>>> print(bin(15))
```

```
0b1111
```

```
>>> print(bin(22))
```

```
0b10110
```

15

0	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---

22

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

15



22



15



22



Two 1

&

1

1

&

15



22



15



22



15



22



6





python3

```
>>> print(15 & 22)
```



python3

```
>>> print(15 & 22)  
6
```



python3

```
>>> print(15 | 22)
```

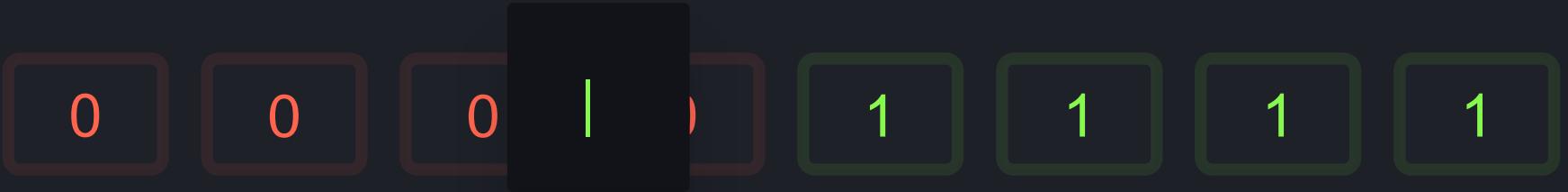
15



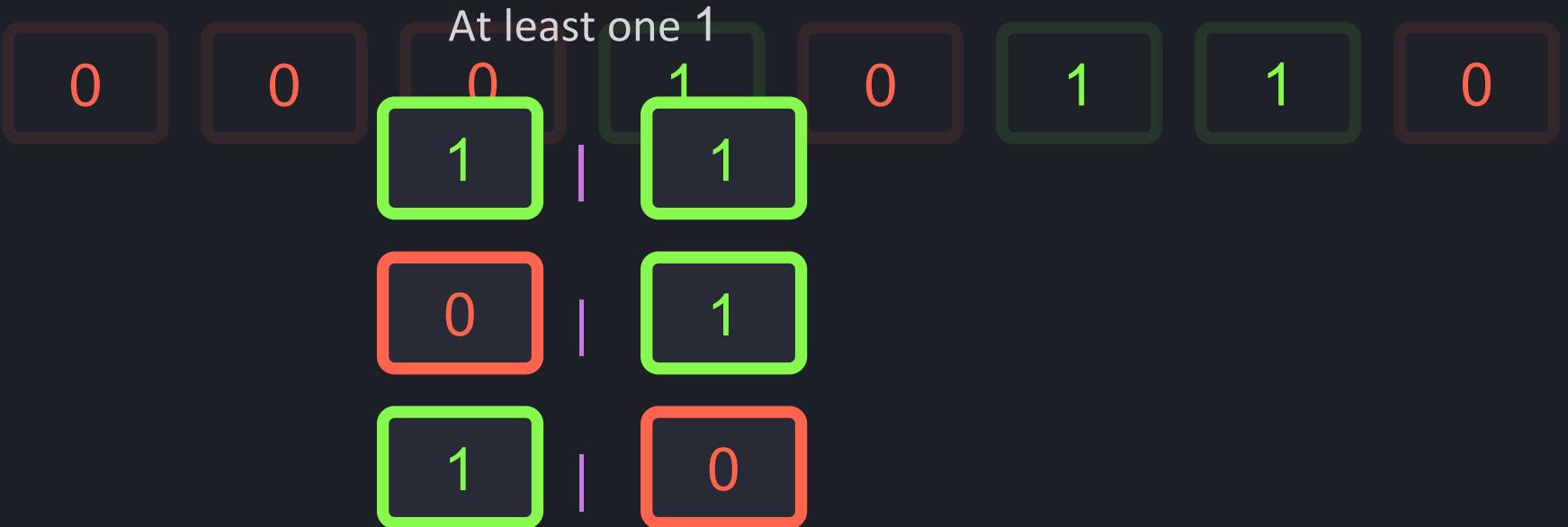
22



15



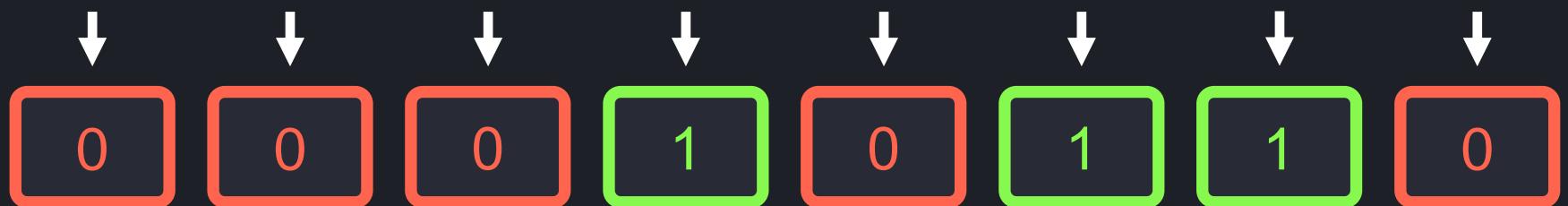
22



15



22



15



22



31





python3

```
>>> print(15 | 22)
```



python3

```
>>> print(15 | 22)
31
```



python3

```
>>> print(15 ^ 22)
```

15



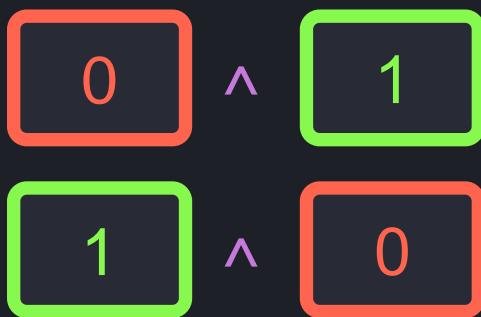
22



15



22



15



22



15



22



25





python3

```
>>> print(~22)
```

22



22



-23



Without Shortcut Operator

```
bit1 = bit1 & 22
```

```
bit1 = bit1 | 22
```

```
bit1 = bit1 ^ 22
```

With Shortcut Operator

```
bit1 &= 22
```

```
bit1 |= 22
```

```
bit1 ^= 22
```

Bit Shifting

>>

<<

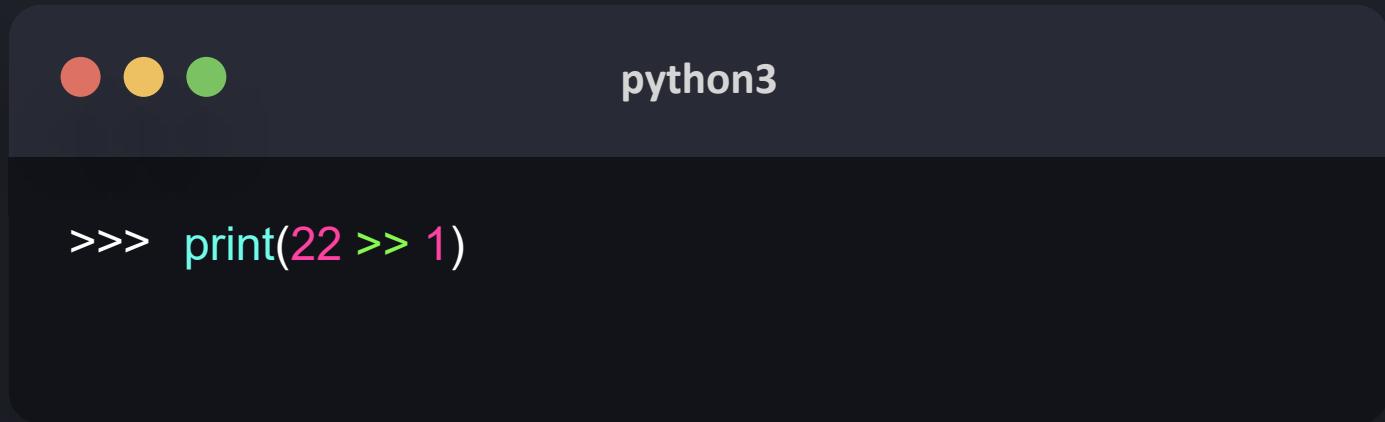
Bit Shift Right

Bit Shift Left



python3

```
>>> print(22 >> 1)
```



A screenshot of a macOS terminal window. The window title is "python3". The terminal prompt is ">>>". The user has typed the Python expression `print(22 >> 1)`. The terminal is dark-themed.

22

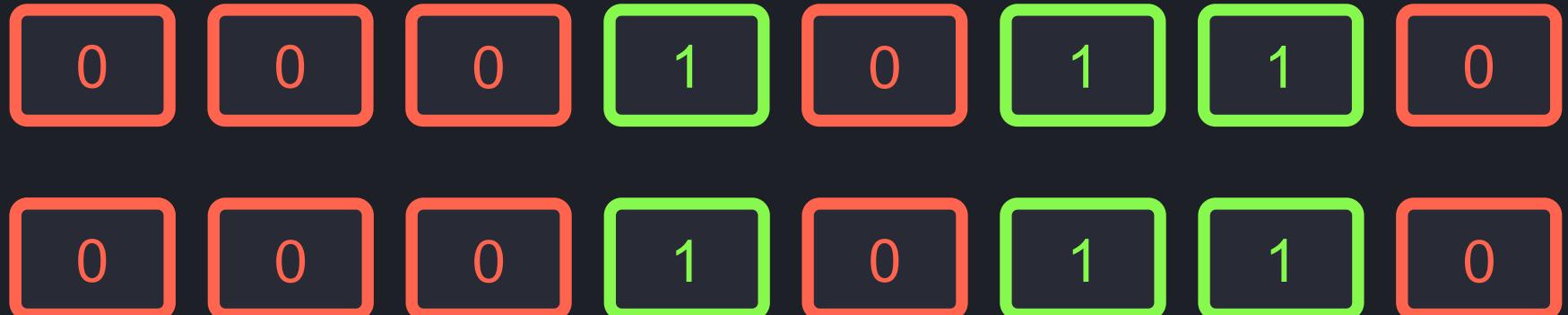
0 0 0 1 0 1 1 0



python3

```
>>> print(22 >> 1)
```

22





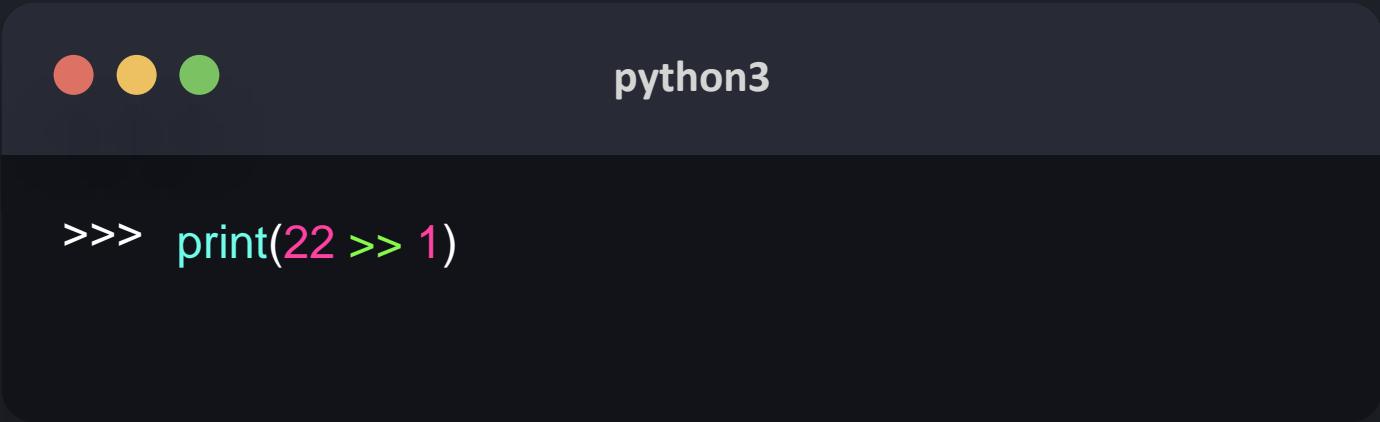
python3

```
>>> print(22 >> 1)
```

22

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

0	0	0	0	1	0	1	1
---	---	---	---	---	---	---	---



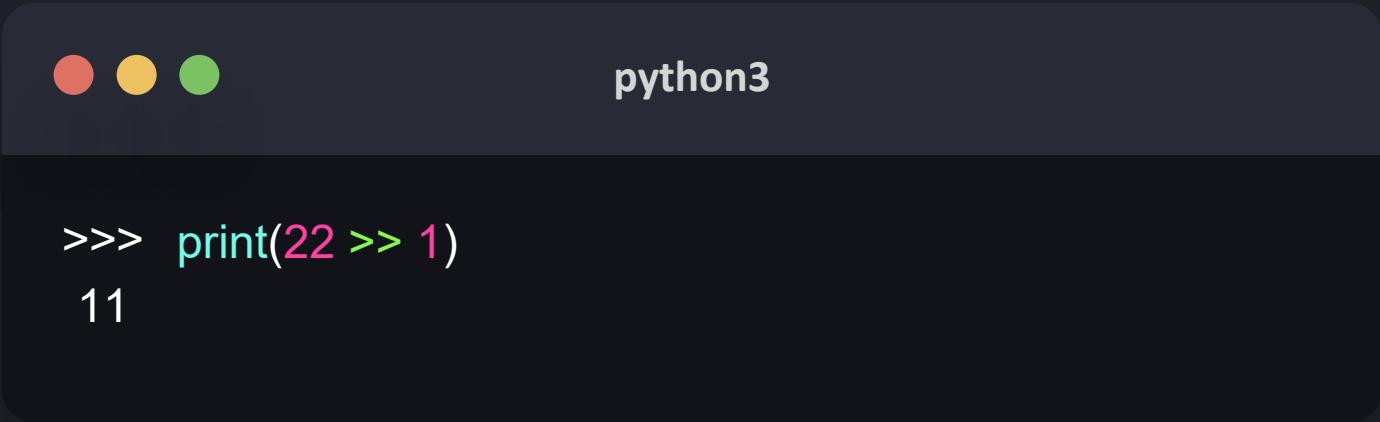
A screenshot of a macOS terminal window. The window title is "python3". The terminal prompt is ">>>". The user has entered the Python code "print(22 >> 1)".

22

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

11

0	0	0	0	1	0	1	1
---	---	---	---	---	---	---	---



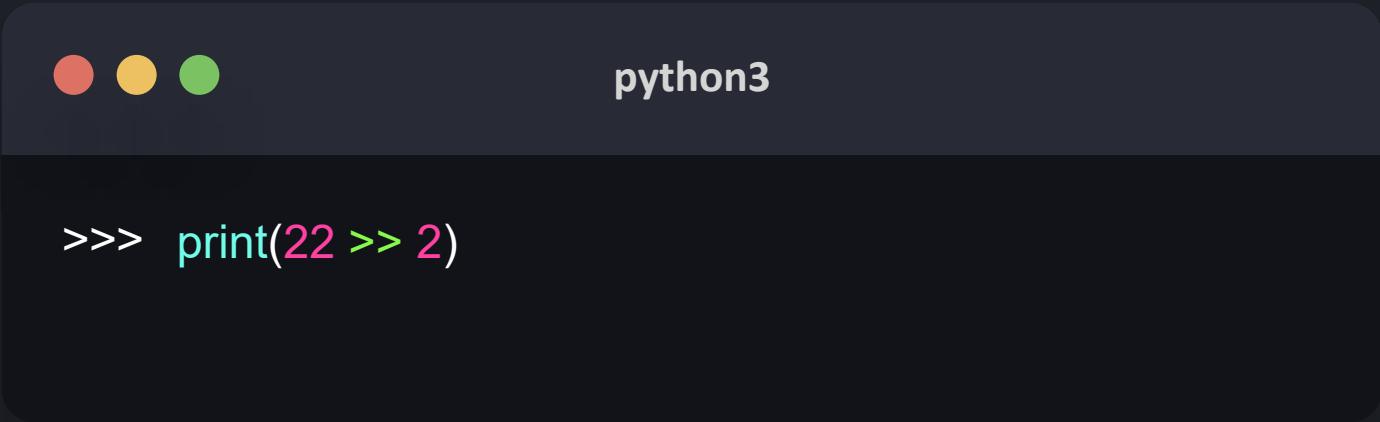
```
python3
>>> print(22 >> 1)
11
```

22

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

11

0	0	0	0	1	0	1	1
---	---	---	---	---	---	---	---

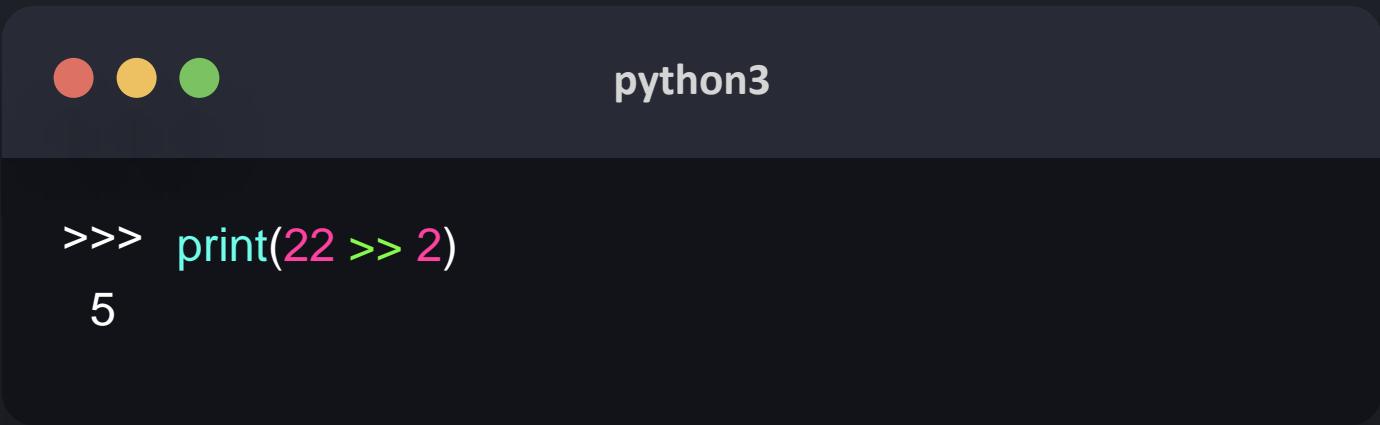


A screenshot of a macOS terminal window. The window title is "python3". The terminal prompt is ">>>". The user has typed the Python expression `print(22 >> 2)`. The terminal is dark-themed.

22

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---



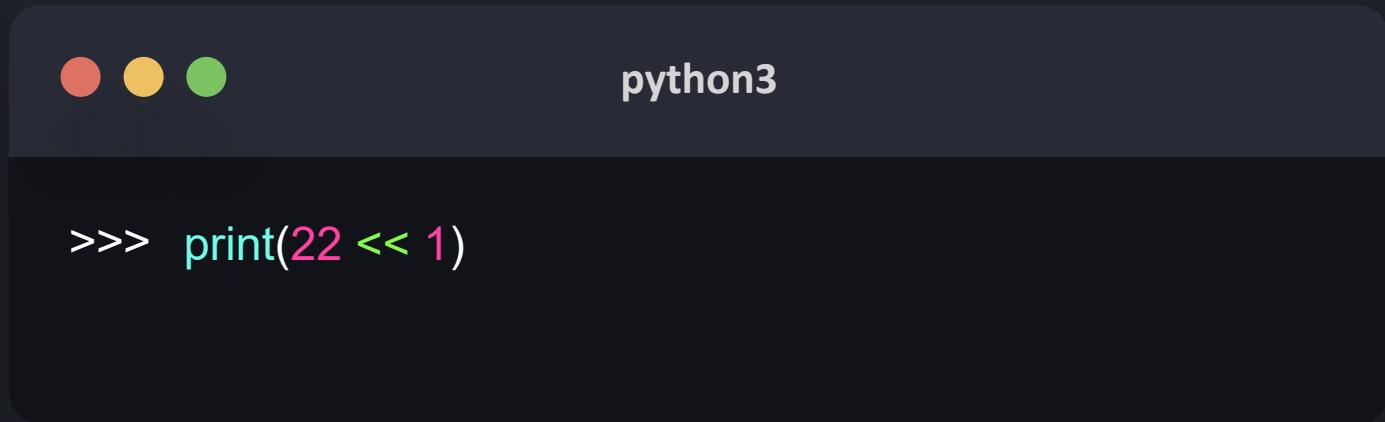
```
python3
>>> print(22 >> 2)
5
```

22

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

5

0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---



A screenshot of a macOS terminal window. The window title is "python3". The terminal prompt is ">>>". The command entered is "print(22 << 1)".

22

0 0 0 1 0 1 1 0



python3

```
>>> print(22 << 1)
```

22

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---



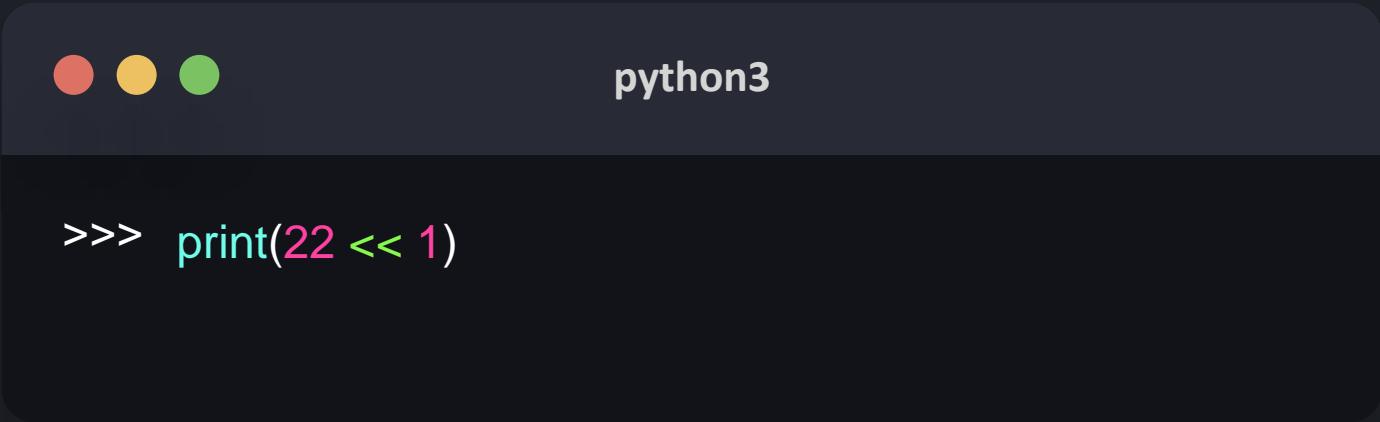
python3

```
>>> print(22 << 1)
```

22

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

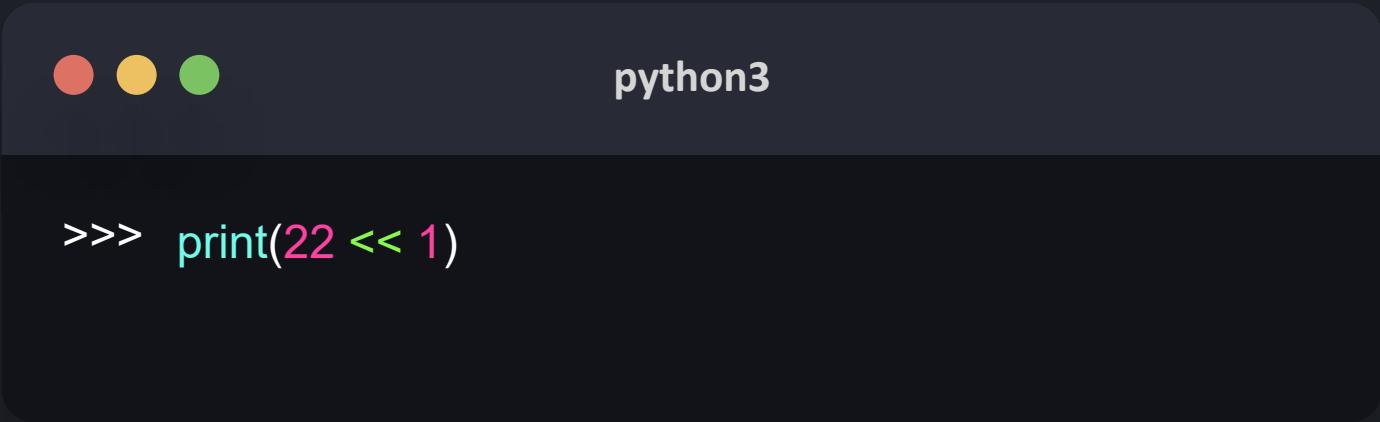


```
python3
>>> print(22 << 1)
```

22

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

0	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---



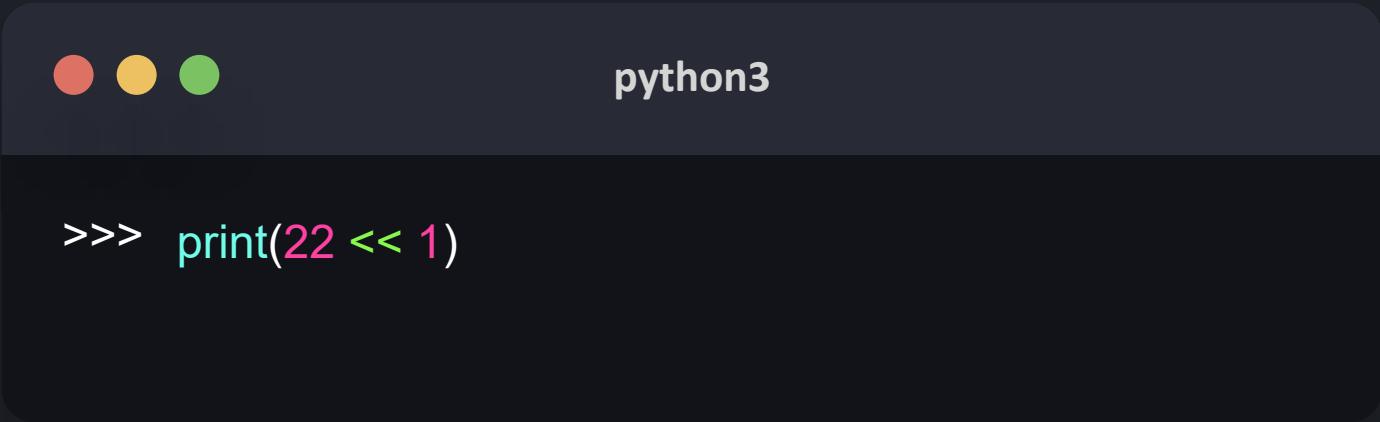
```
python3
>>> print(22 << 1)
```

22

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

44

0	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---



```
python3
>>> print(22 << 1)
```

22

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

44

0	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---

```
print(22 // 2)
```



```
print(22 >> 1)
```

```
print(22 // 4)
```



```
print(22 >> 2)
```

```
print(22 * 2)
```



```
print(22 << 1)
```

```
print(22 * 4)
```



```
print(22 << 2)
```

Operators

- Logical operators `and` `not` and `or` return boolean values based on the passed values
- Bitwise operators `&` `|` `^` and `~` allow us to manipulate single bits of data, and return `0` or `1` based on the value of the bits that are used
- Bit shifting can be done with the `<<` and `>>` operators



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Lists



python3

```
>>> countries = ["USA", "Canada", "India"]
```

0

"USA"

1

"Canada"

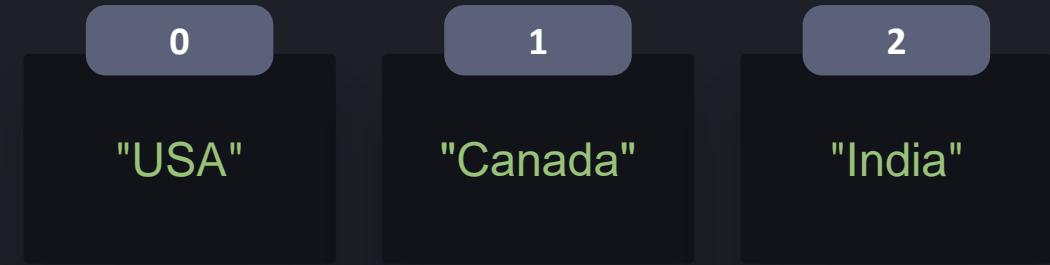
2

"India"



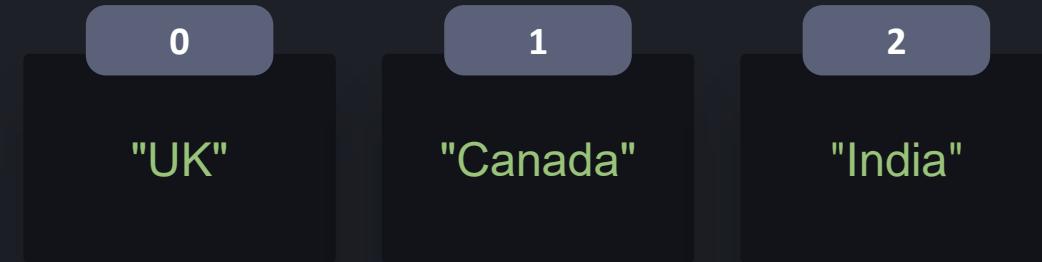
python3

```
>>> countries = ["USA", "Canada", "India"]
>US print(countries[0])
>A> print(countries[1])
Canada
>>> print(countries[2])
India
```



A screenshot of a terminal window. The title bar says "python3". The window contains the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> countries[0] = "UK"
```



A screenshot of a macOS terminal window titled "python3". The window has a dark theme with red, yellow, and green window control buttons at the top-left. The title bar displays "python3". The main pane shows Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> countries[0] = "UK"
```

len()



python3

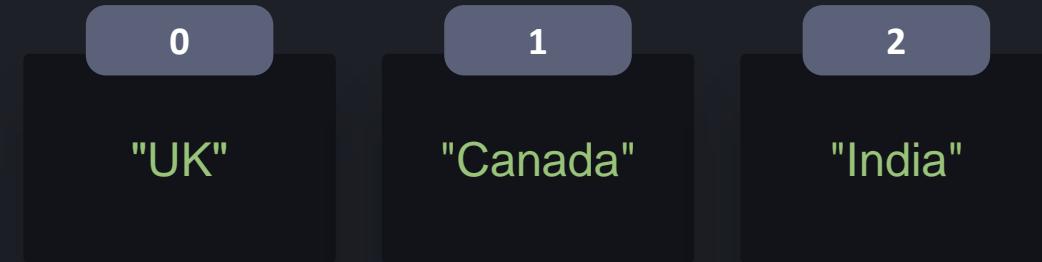
```
>>> countries = ["USA", "Canada", "India"]
>>> len(countries)
```

len()



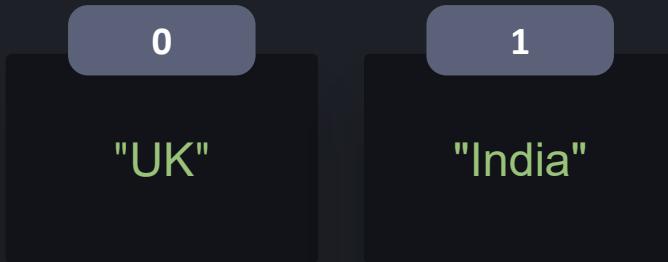
python3

```
>>> countries = ["USA", "Canada", "India"]
>>> len(countries)
3
```



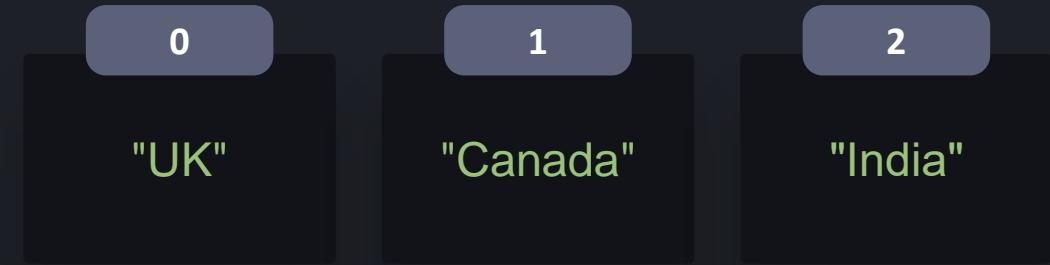
A screenshot of a macOS terminal window titled 'python3'. The window has a dark theme with red, yellow, and green window control buttons. The title bar shows the window name 'python3'. The terminal window displays the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> del countries[1]
```



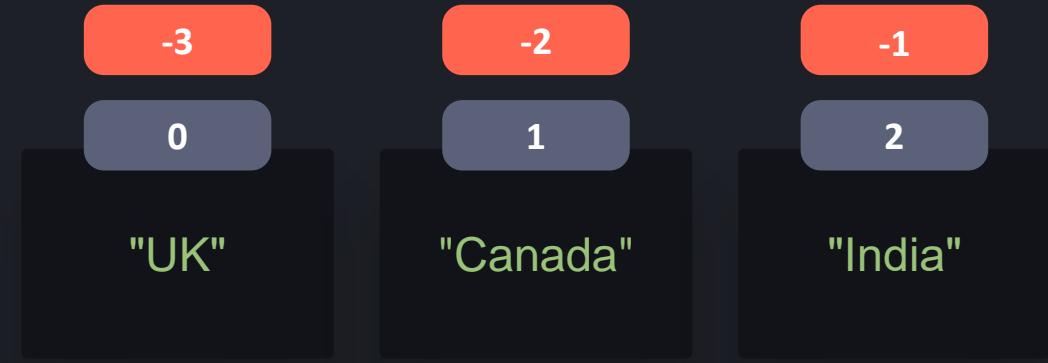
A screenshot of a macOS terminal window. The window title is "python3". The terminal interface includes three colored window control buttons (red, yellow, green) on the left. The main area displays the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> del countries[1]
```



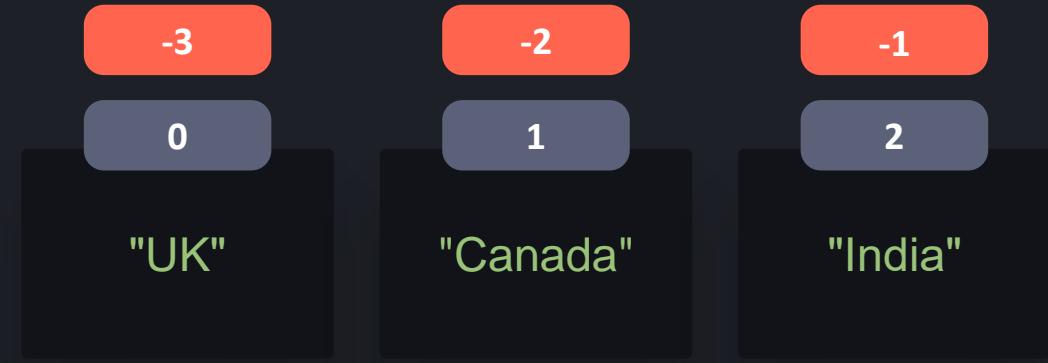
A screenshot of a terminal window titled "python3". The window has a dark theme with three colored window control buttons (red, yellow, green) at the top-left. The title bar displays the text "python3". The main area of the terminal shows the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
```



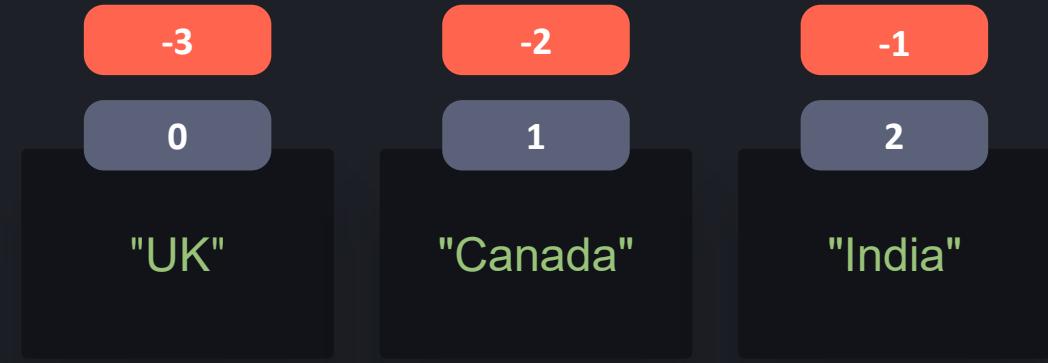
A screenshot of a terminal window titled "python3". The window has a dark theme with three colored window control buttons (red, yellow, green) at the top-left. The title bar contains the text "python3". The main area of the terminal shows the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
```



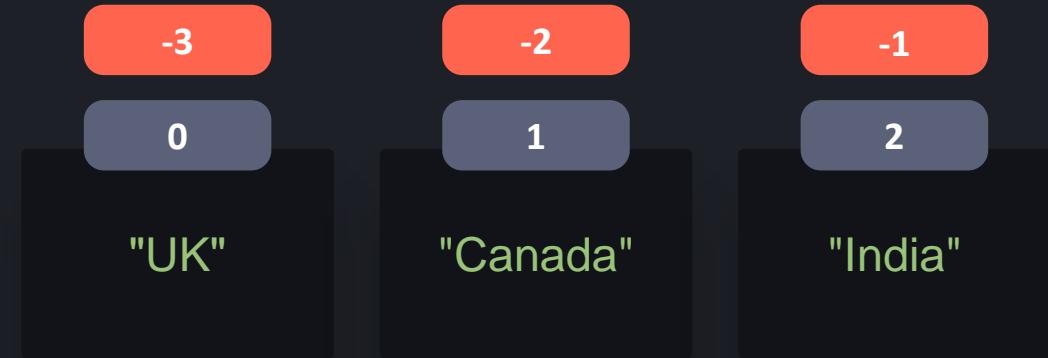
A screenshot of a macOS terminal window titled "python3". The window has a dark theme with red, yellow, and green window control buttons. The terminal displays the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> print(countries[-1])
```



A screenshot of a terminal window titled "python3". The window has a dark theme with three small colored circles (red, yellow, green) in the top-left corner. The terminal displays the following Python code and output:

```
>>> countries = ["USA", "Canada", "India"]
>>> print(countries[-1])
"India"
```



A screenshot of a terminal window titled "python3". The window has a dark theme with three small colored circles (red, yellow, green) in the top-left corner. The terminal output shows:

```
>>> countries = ["USA", "Canada", "India"]
>>> print(countries[4])
IndexError: list index out of range
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Lists - Methods

list.append()

list.insert()

Functions

`print()`

`len()`

`input()`

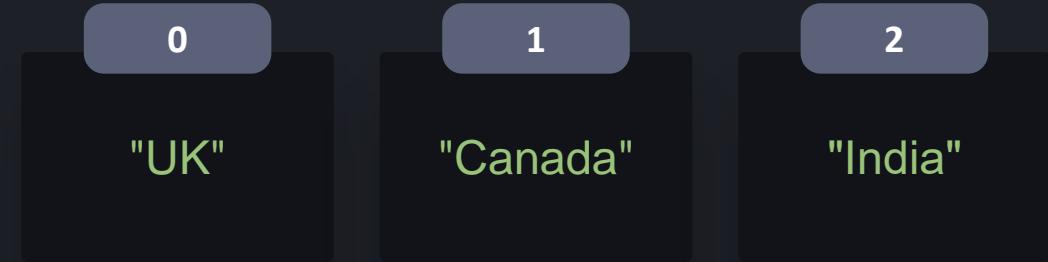
Methods

`list.append()`

`list.insert()`

list.append()

list.insert()



A screenshot of a terminal window titled "python3". The window has a dark theme with three small colored circles (red, yellow, green) in the top-left corner. The terminal displays the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> countries.append("Spain")
```

0

"UK"

1

"Canada"

2

"India"

3

"Spain"



python3

```
>>> countries = ["USA", "Canada", "India"]
>>> countries.append("Spain")
```

0

"UK"

1

"Canada"

2

"India"

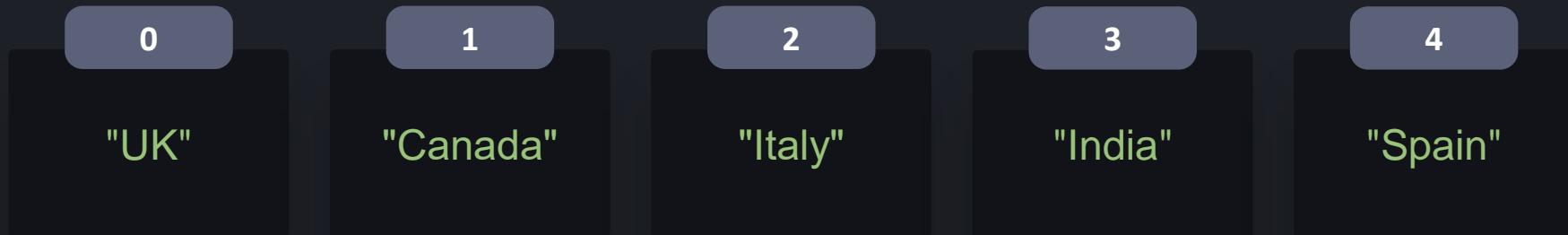
3

"Spain"

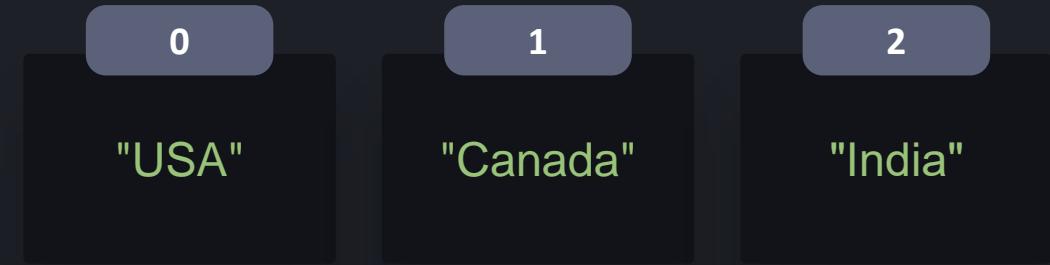


python3

```
>>> countries = ["USA", "Canada", "India"]
>>> countries.append("Spain")
>>> countries.insert(2, "Italy")
```

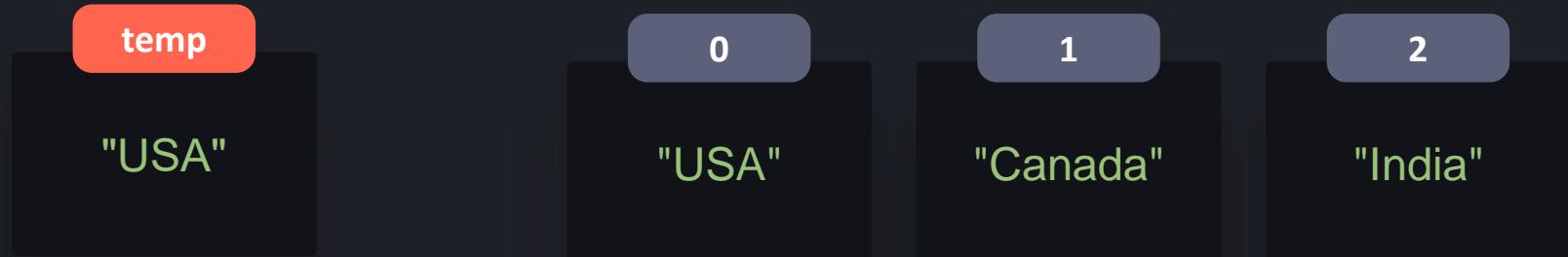


```
>>> countries = ["USA", "Canada", "India"]
>>> countries.append("Spain")
>>> countries.insert(2, "Italy")
```



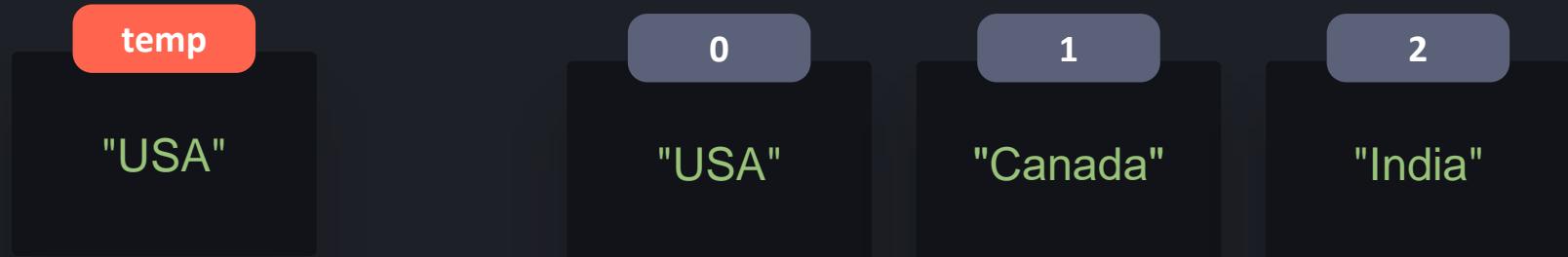
A screenshot of a terminal window titled "python3". The window has a dark background with a title bar featuring three colored circles (red, yellow, green) on the left and the text "python3" on the right. The main area of the terminal shows a single line of code:

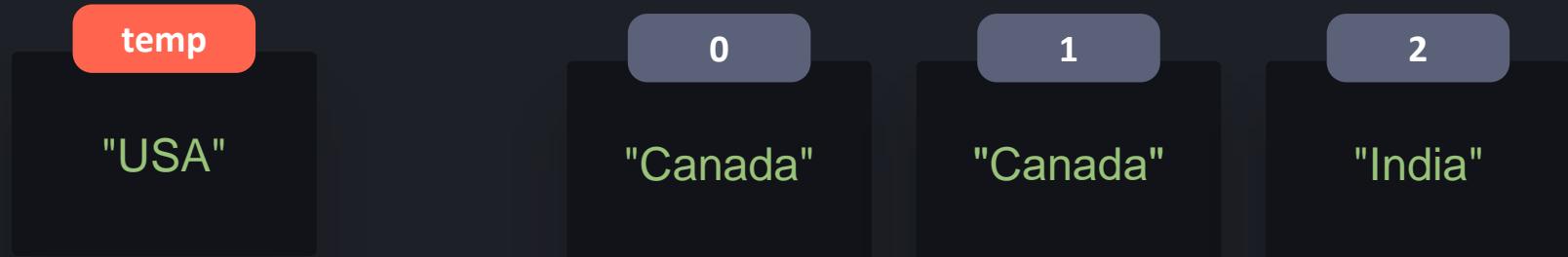
```
>>> countries = ["USA", "Canada", "India"]
```



A screenshot of a macOS terminal window titled 'python3'. The window has a dark theme with three colored window control buttons (red, yellow, green) in the top-left corner. The title bar shows the text 'python3'. The main pane displays the following Python code:

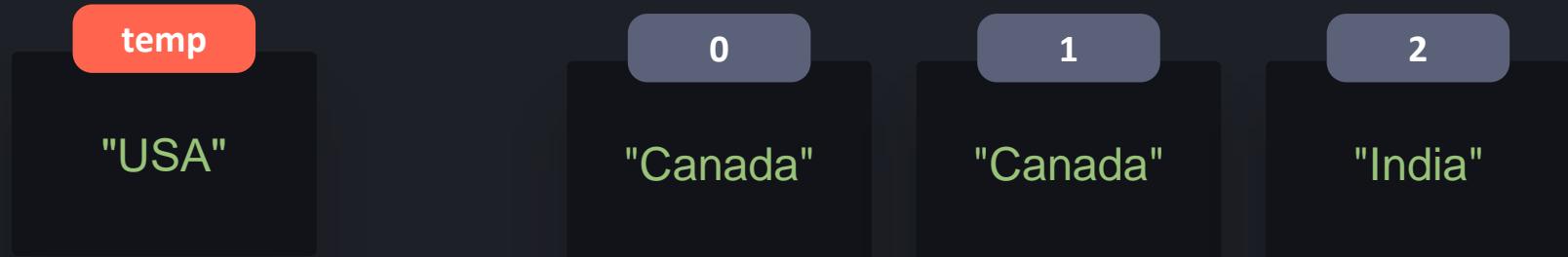
```
>>> countries = ["USA", "Canada", "India"]
>>> temp = countries[0]
```





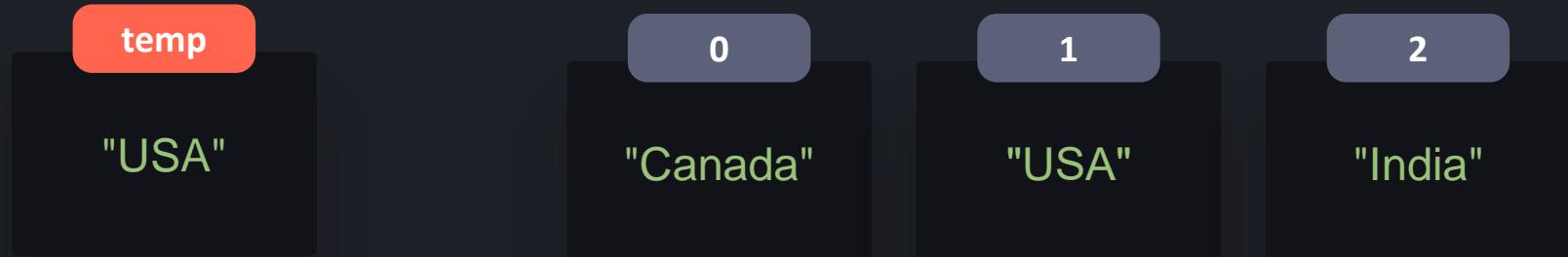
A screenshot of a macOS terminal window titled "python3". The window shows the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> temp = countries[0]
>>> countries[0] = countries[1]
```



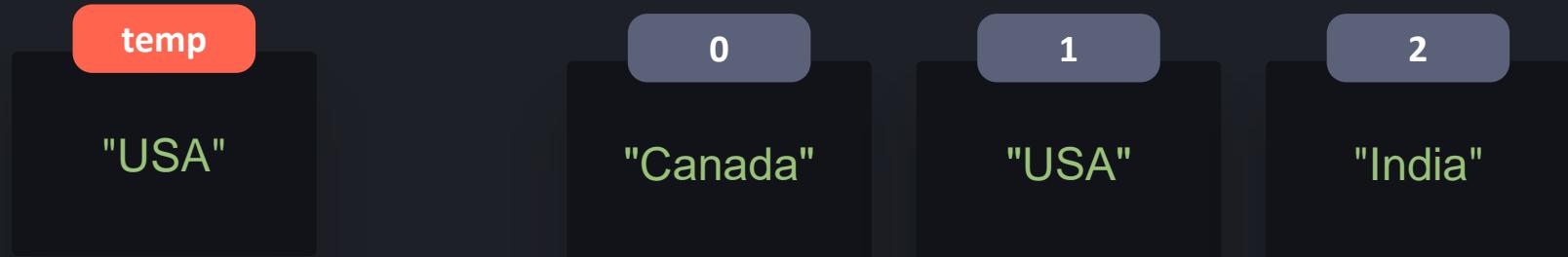
A terminal window titled `python3` is shown, displaying the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> temp = countries[0]
>>> countries[0] = countries[1]
>>> countries[1] = temp
```



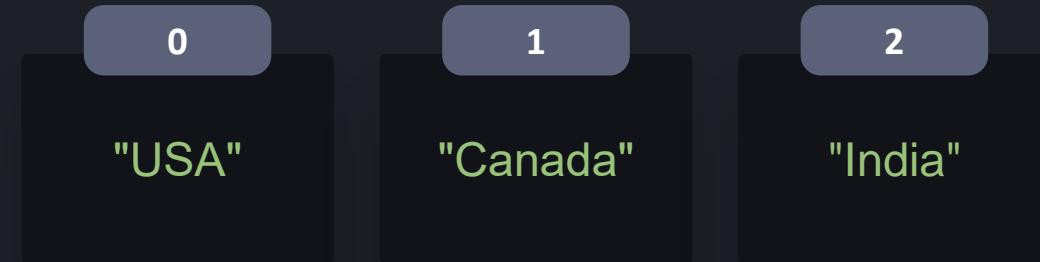
A screenshot of a terminal window titled `python3`. The window has three colored window control buttons (red, yellow, green) at the top left. The terminal displays the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> temp = countries[0]
>>> countries[0] = countries[1]
>>> countries[1] = temp
```



A terminal window titled `python3` is shown executing the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> temp = countries[0]
>>> countries[0] = countries[1]
>>> countries[1] = temp
```



A screenshot of a terminal window titled "python3". The window has a dark theme with red, yellow, and green window control buttons. The terminal displays the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> countries[0], countries[1] = countries[1], countries[0]
```

0

1

2

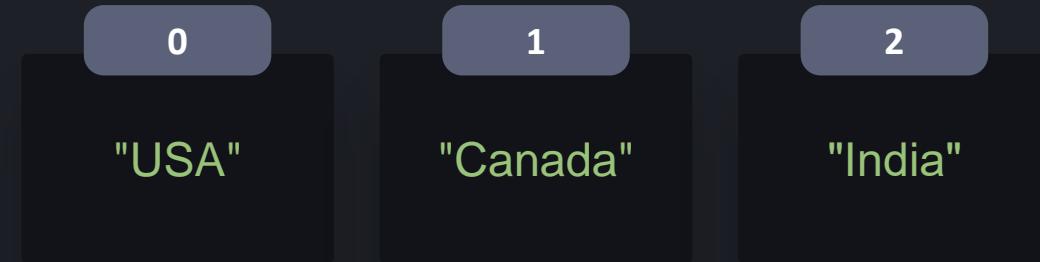
"USA"

"Canada"

"India"

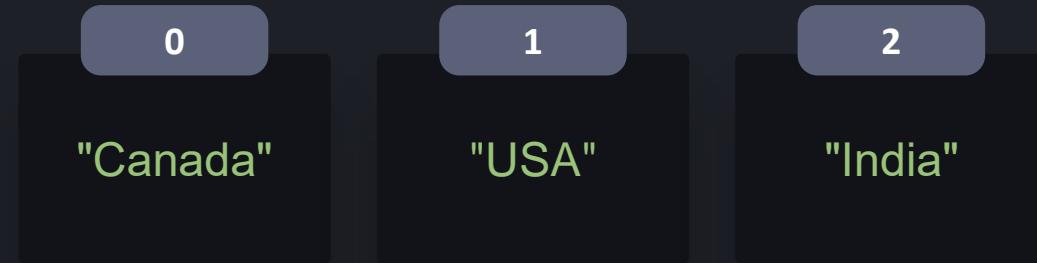
countries[0], countries[1] = countries[1], countries[0]

~~>>> countries = ["USA", "Canada", "India"]~~



A screenshot of a terminal window titled "python3". The window has a dark background with three colored window control buttons (red, yellow, green) on the top-left. The title bar contains the text "python3". The terminal window displays the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> countries[0], countries[1] = countries[1], countries[0]
```



A screenshot of a terminal window titled "python3". The window has a dark theme with three colored window control buttons (red, yellow, green) at the top-left. The title bar contains the text "python3". The main area of the terminal shows the following Python code:

```
>>> countries = ["USA", "Canada", "India"]
>>> countries[0], countries[1] = countries[1], countries[0]
```

list.sort()

list.reverse()

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> ages.sort()
```

0

24

1

46

2

56

3

72



python3

```
>>> ages = [56, 72, 24, 46]
>>> ages.sort()
>>> print(ages)
```

0

24

1

46

2

56

3

72



python3

```
>>> ages = [56, 72, 24, 46]
>>> ages.sort()
>>> print(ages)
[24, 46, 56, 72]
```

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> ages.reverse()
```



A screenshot of a terminal window titled "python3". The window has three small colored icons (red, yellow, green) in the top-left corner. The title bar contains the text "python3". The main area of the terminal shows the following Python code being run:

```
>>> ages = [56, 72, 24, 46]
>>> ages.reverse()
>>> print(ages)
```

0

46

1

24

2

72

3

56



python3

```
>>> ages = [56, 72, 24, 46]
>>> ages.reverse()
>>> print(ages)
[46, 24, 72, 56]
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Iterating Lists

0

56

1

72

2

24

3

46



total

0

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
```

total

0

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

0

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

0

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

56

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

56

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

128

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

0

1

2

3

128

56

72

24

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

0

1

2

3

152

56

72

24

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

0

1

2

3

152

56

72

24

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

0

1

2

3

198

56

72

24

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

198

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

198

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
    total += age
```

total

198

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
...     total += age
>>> average = total / len(ages)
```

total

198

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
...     total += age
>>> average = total / len(ages)
>>> print(average)
```

total

198

0

56

1

72

2

24

3

46



python3

```
>>> ages = [56, 72, 24, 46]
>>> total = 0
>>> for age in ages:
...     total += age
>>> average = total / len(ages)
>>> print(average)
49.5
```

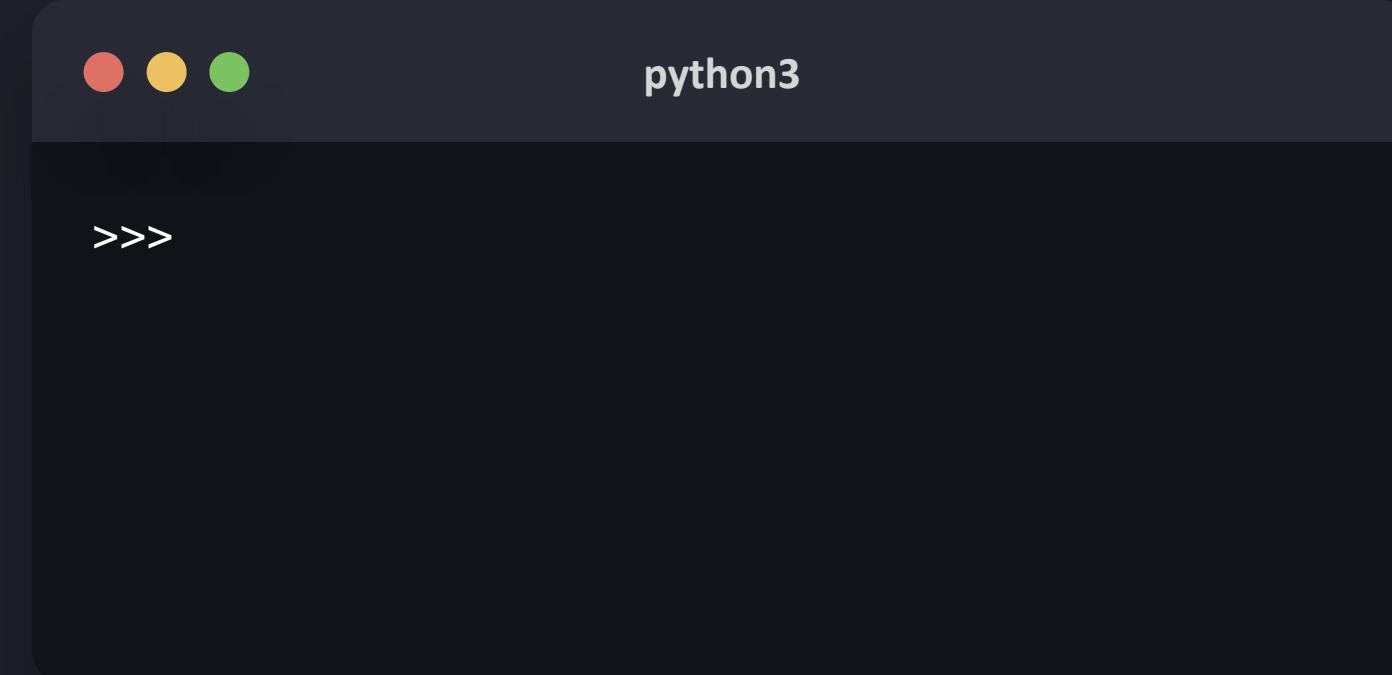


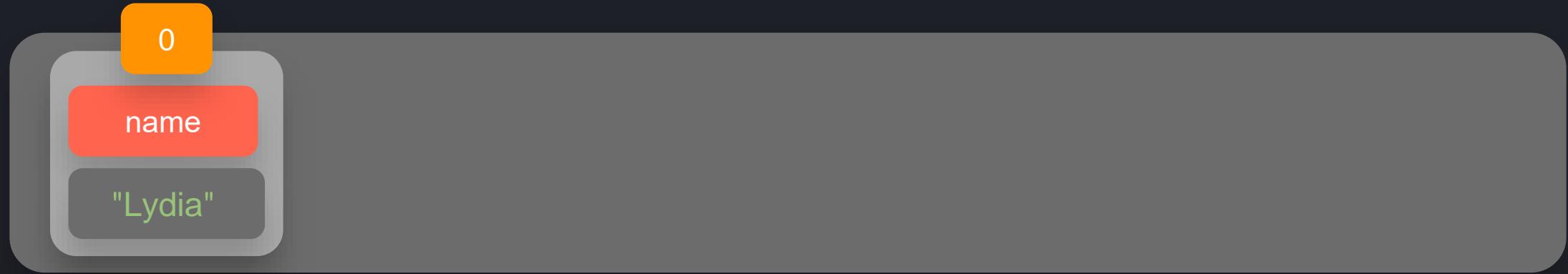
KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>

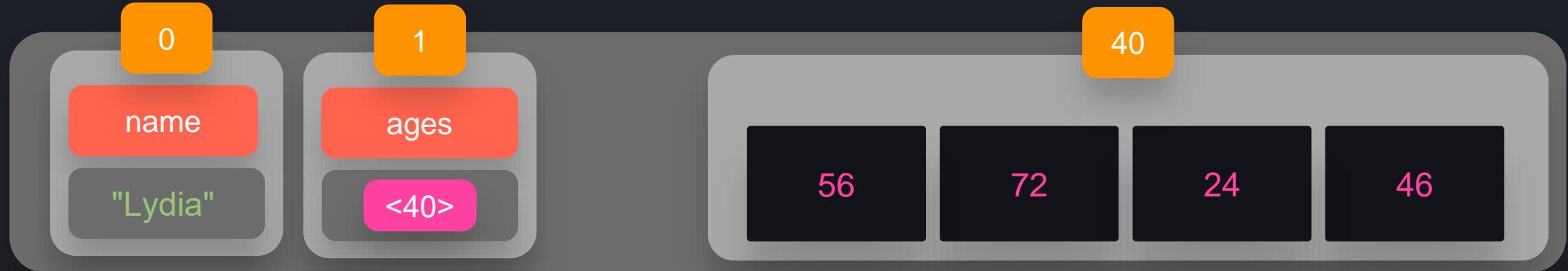


Understanding Lists



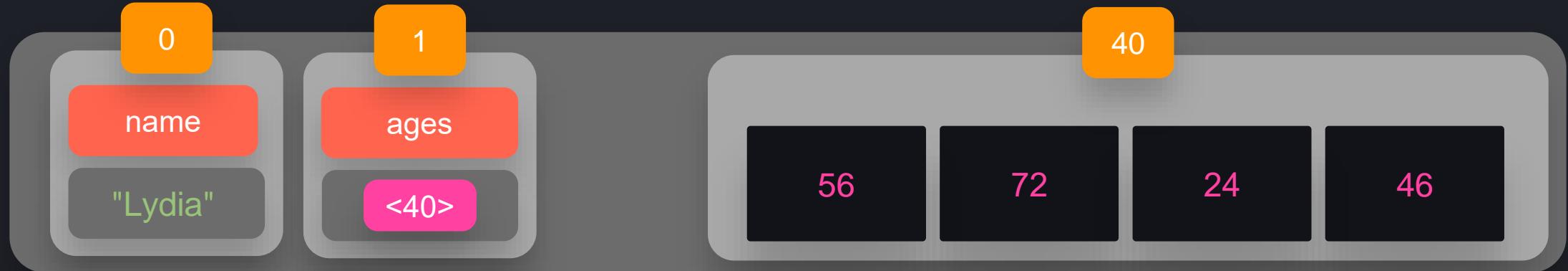


A screenshot of a terminal window titled 'python3'. The window has three colored dots (red, yellow, green) in the top-left corner. The text 'python3' is displayed in the title bar. In the main area, the command `>>> name = "Lydia"` is shown in white text on a black background.



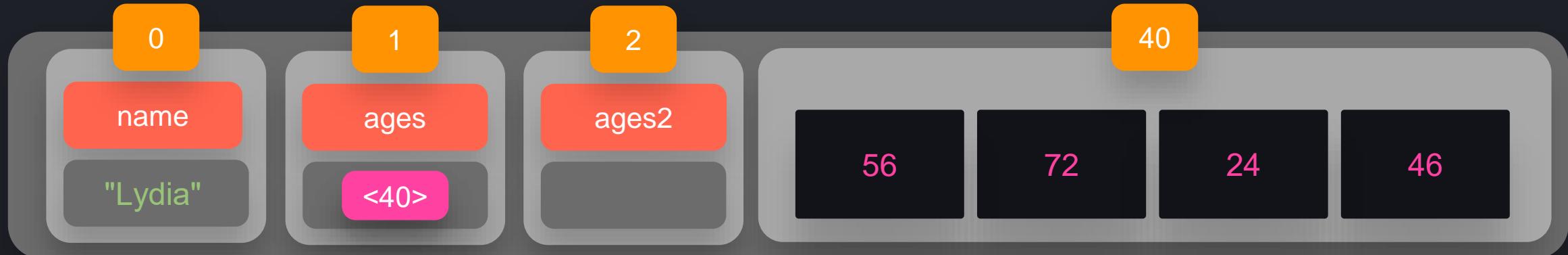
A screenshot of a terminal window titled 'python3'. The window has three colored dots (red, yellow, green) at the top left. The terminal displays the following Python code:

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
```



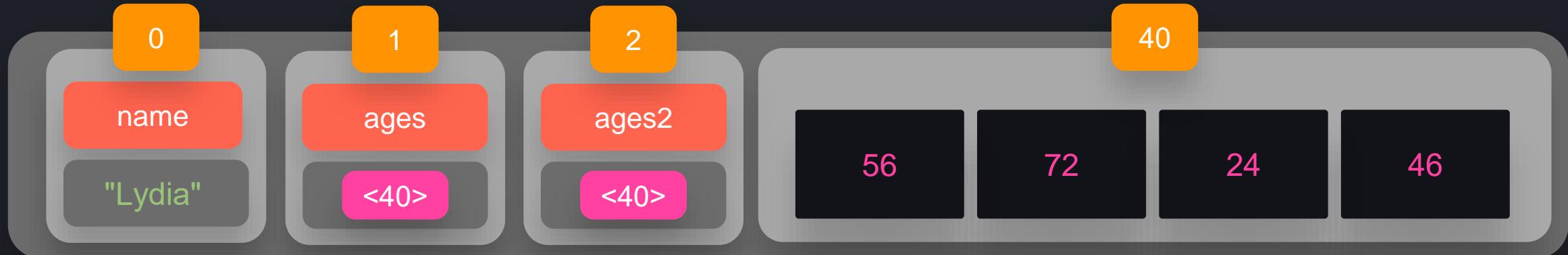
python3

```
>>> name = "Lydia"  
>>> ages = [56, 72, 24, 46]  
>>> ages2 = ages
```



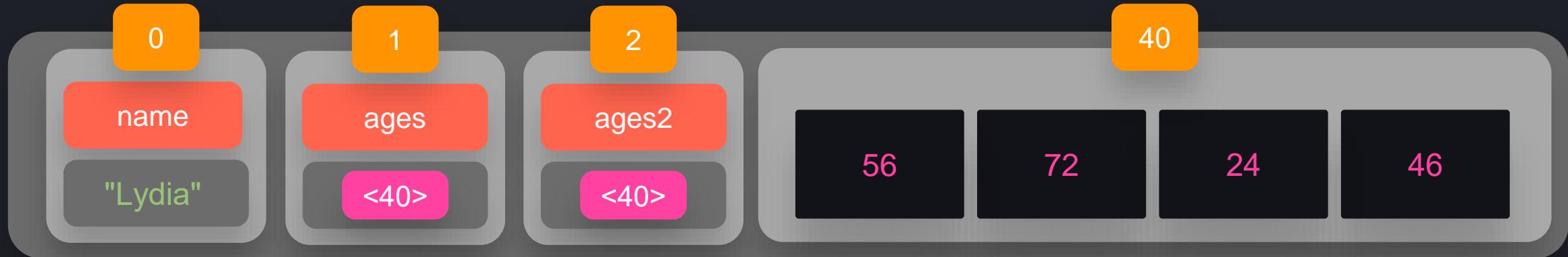
python3

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
>>> ages2 = ages
```



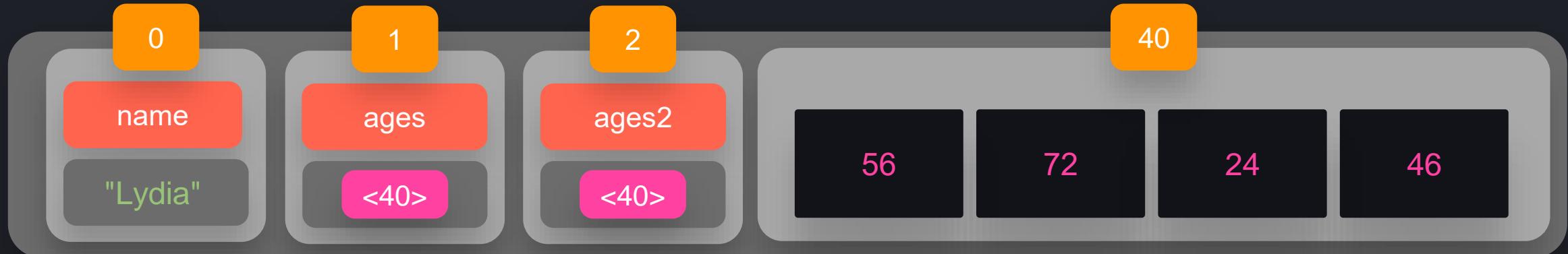
python3

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
>>> ages2 = ages
```



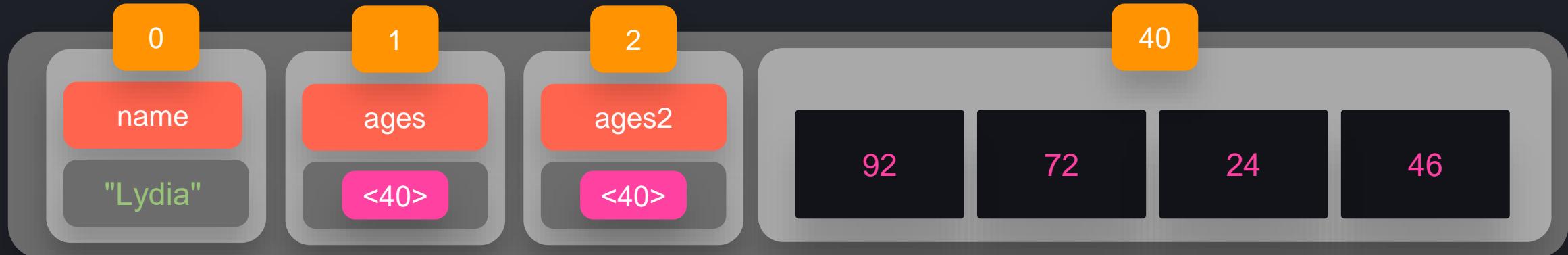
python3

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
>>> ages2 = ages
```



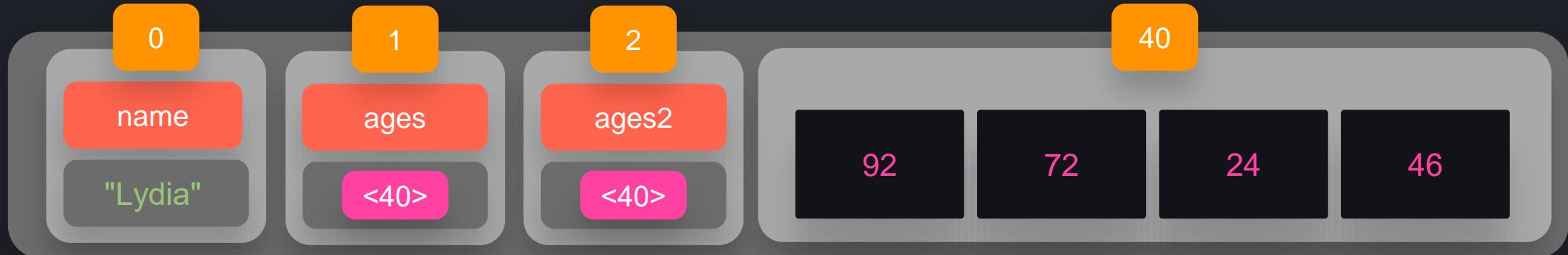
python3

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
>>> ages2 = ages
>>> ages[0] = 92
```



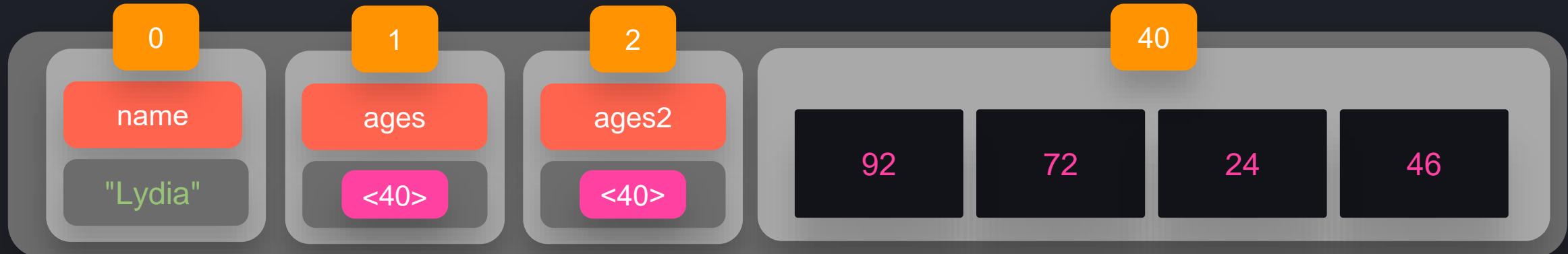
python3

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
>>> ages2 = ages
>>> ages[0] = 92
```



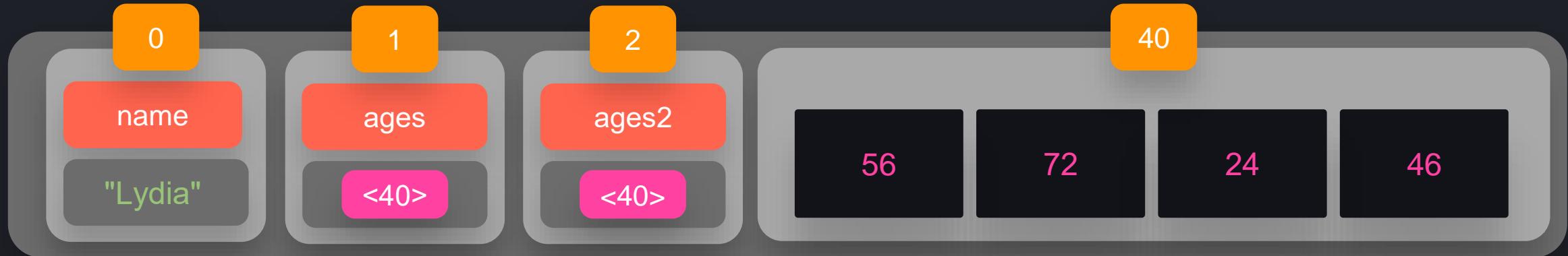
python3

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
>>> ages2 = ages
>>> ages[0] = 92
>>> print(ages2[0])
```



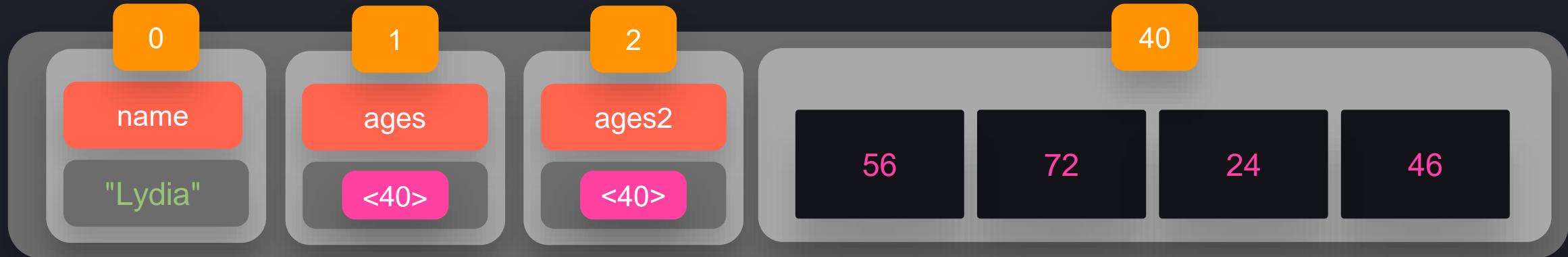
python3

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
>>> ages2 = ages
>>> ages[0] = 92
>>> print(ages2[0])
92
```



python3

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
```



A screenshot of a terminal window titled "python3". The window has three colored dots (red, yellow, green) in the top-left corner. The terminal displays the following Python code:

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
```



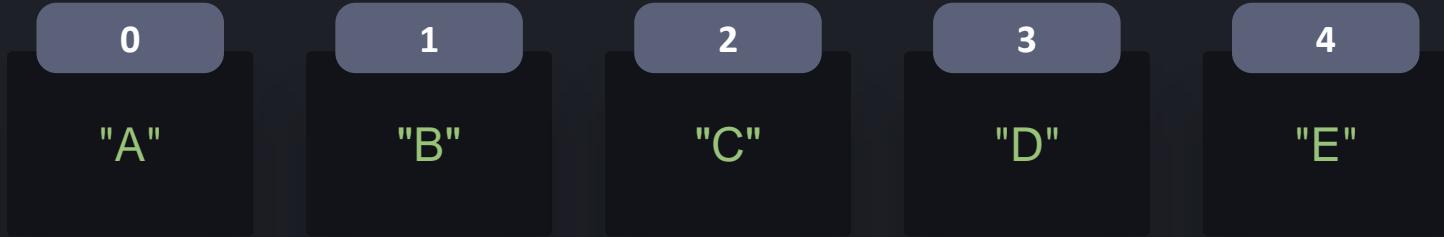
KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



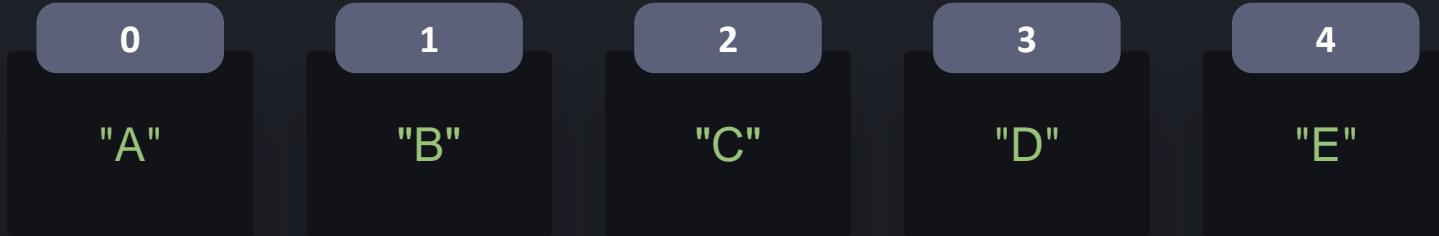
Slicing Lists

list[*start*:*end*]



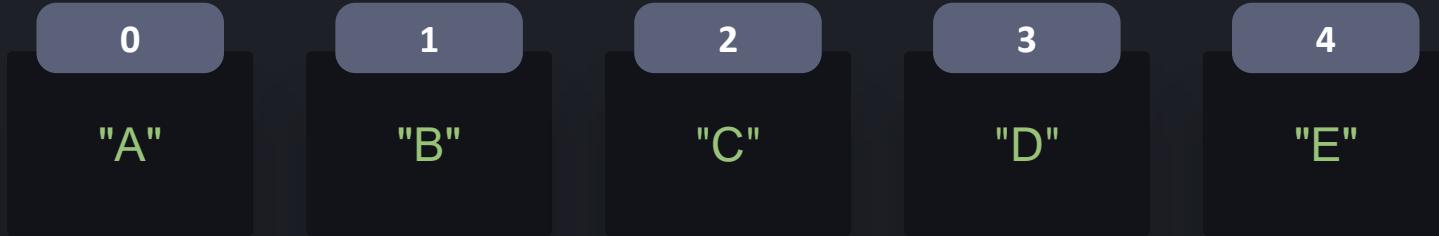
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
```



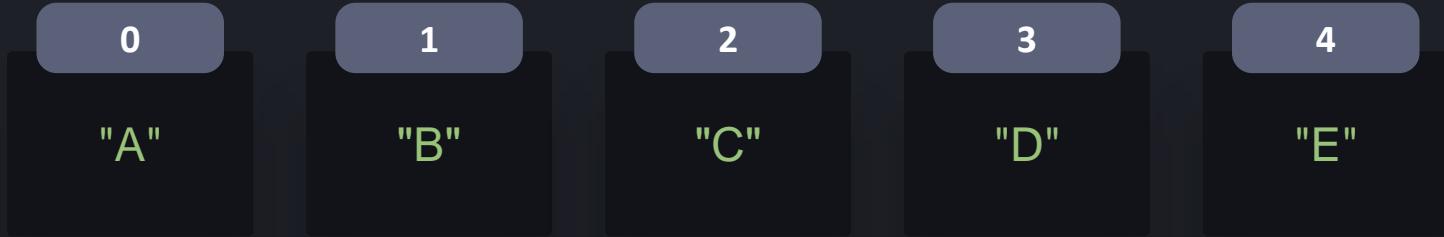
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> firstTwo = letters[0:2]
```



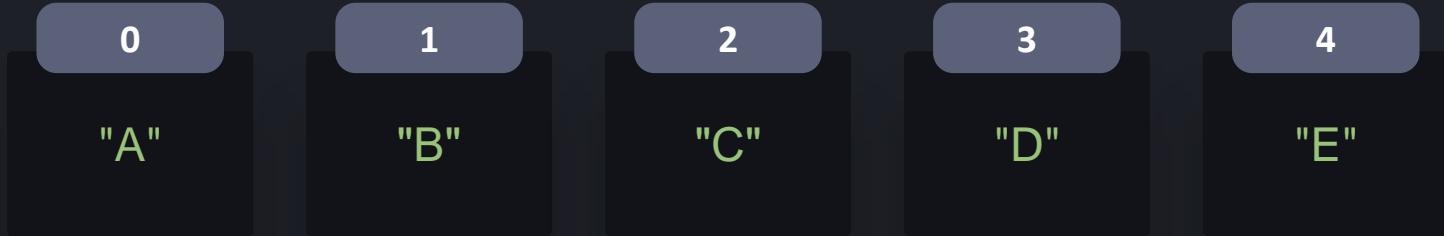
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> firstTwo = letters[0:2]
>>> print(firstTwo)
["A", "B"]
```



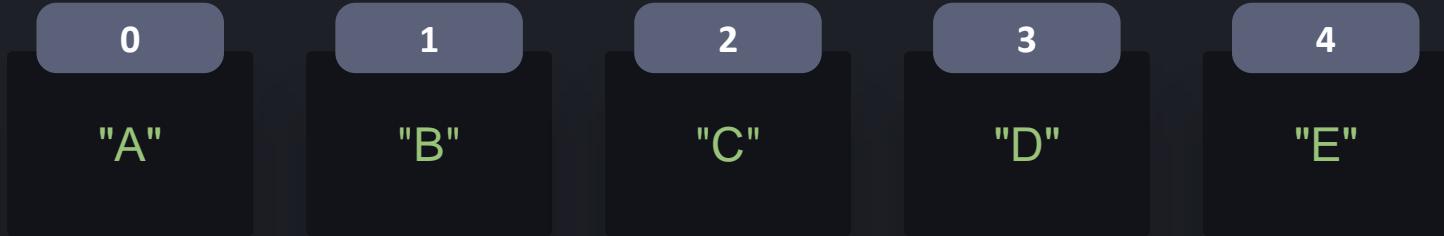
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> firstTwo = letters[0:2]
>>> print(firstTwo)
["A", "B"]
>>> print(letters[1:])
```



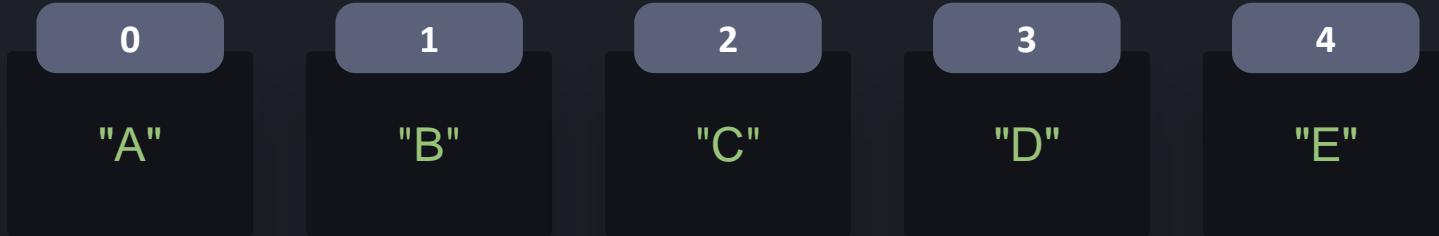
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> firstTwo = letters[0:2]
>>> print(firstTwo)
["A", "B"]
>>> print(letters[1:])
["B", "C", "D", "E"]
```



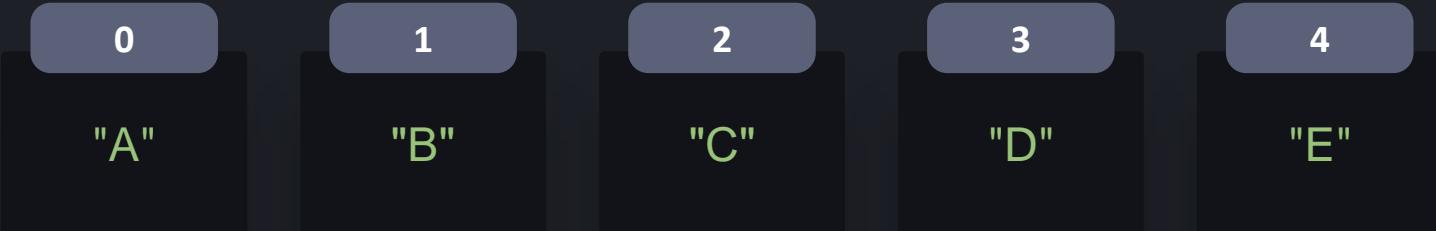
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> firstTwo = letters[0:2]
>>> print(firstTwo)
["A", "B"]
>>> print(letters[1:])
["B", "C", "D", "E"]
>>> print(letters[:3])
```



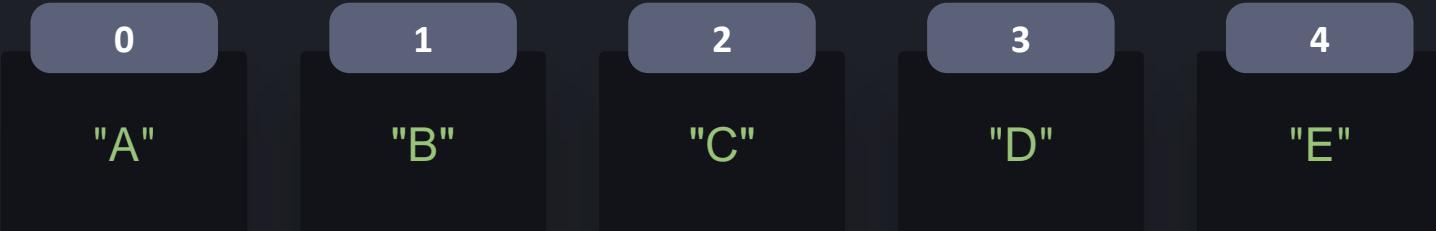
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> firstTwo = letters[0:2]
>>> print(firstTwo)
["A", "B"]
>>> print(letters[1:])
["B", "C", "D", "E"]
>>> print(letters[:3])
["A", "B", "C"]
```



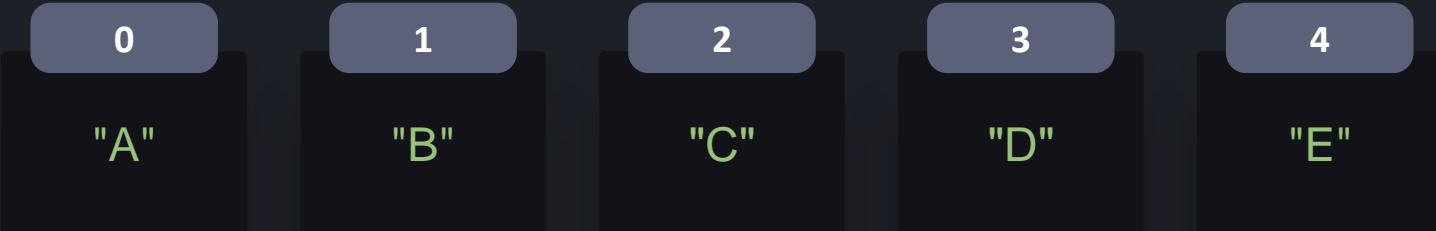
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
```



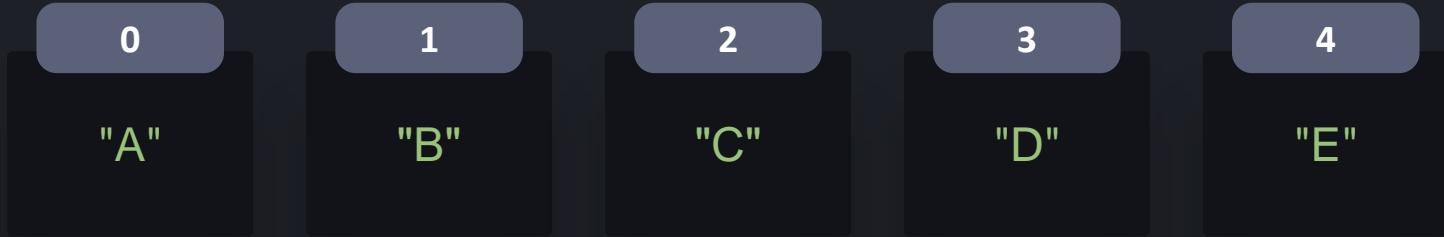
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> print(letters[1:-1])
["B", "C", "D"]
```



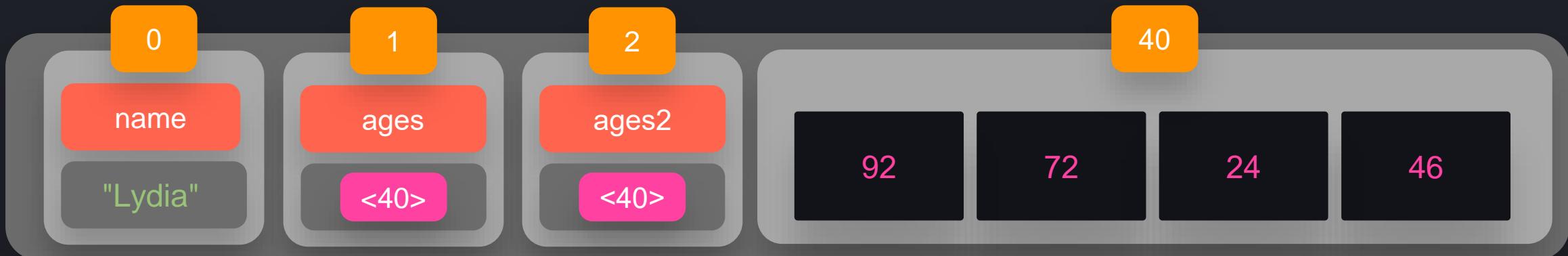
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> print(letters[:])
```



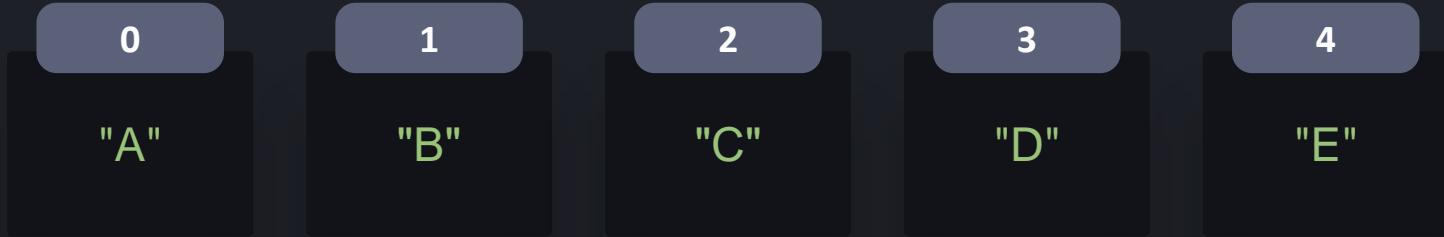
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> print(letters[:])
["A", "B", "C", "D", "E"]
```



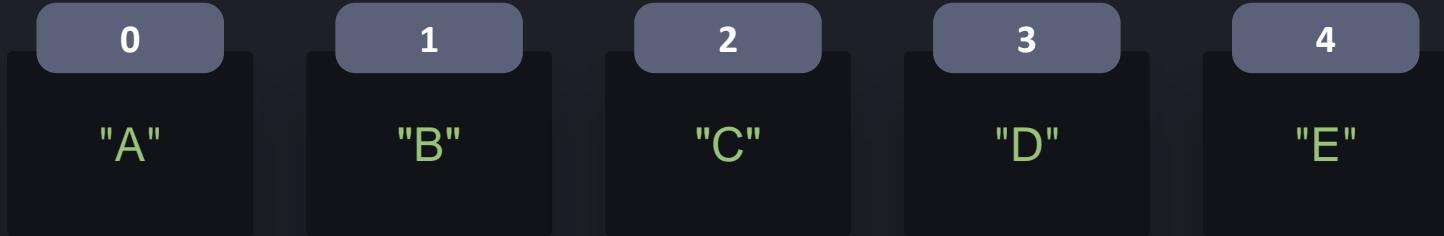
python3

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
>>> ages2 = ages
>>> ages[0] = 92
>>> print(ages2[0])
92
```



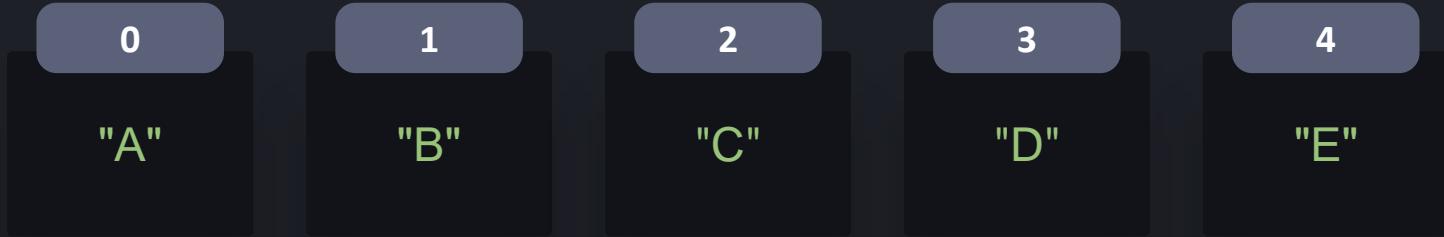
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> print(letters[:])
["A", "B", "C", "D", "E"]
```



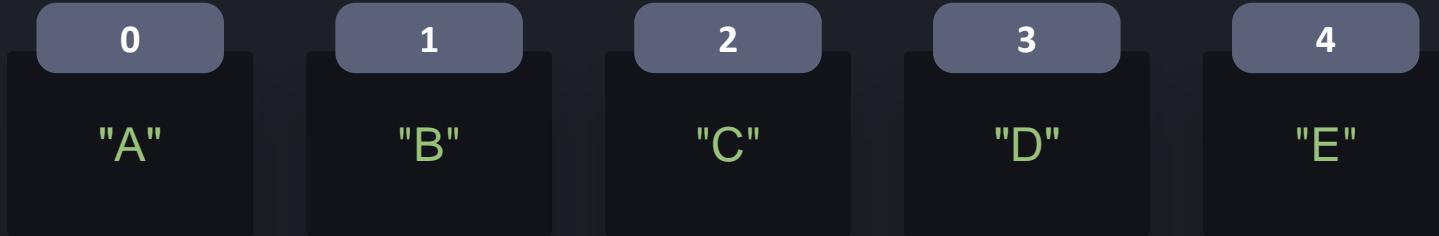
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> del letters[1:3]
```



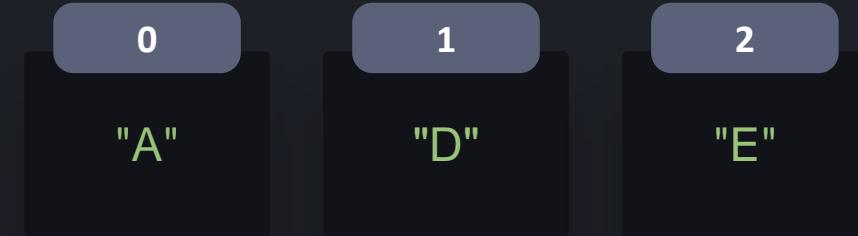
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> del letters[1:3]
```



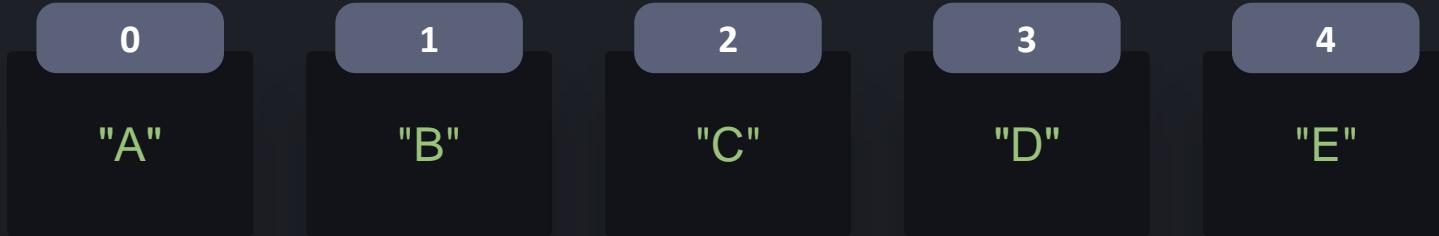
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> del letters[1:3]
```



A screenshot of a terminal window titled "python3". The window has a dark theme with three colored circular icons in the top-left corner (red, yellow, green). The terminal displays the following Python code:

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> del letters[1:3]
```



python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> del letters[:]
```



python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> del letters[:]
>>> print(letters)
```



python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> del letters[:]
>>> print(letters)
[]
```



KodeKloud

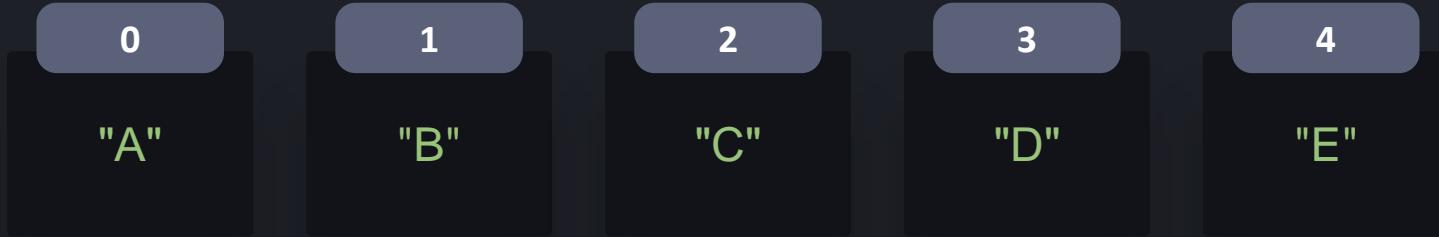
Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Finding in Lists

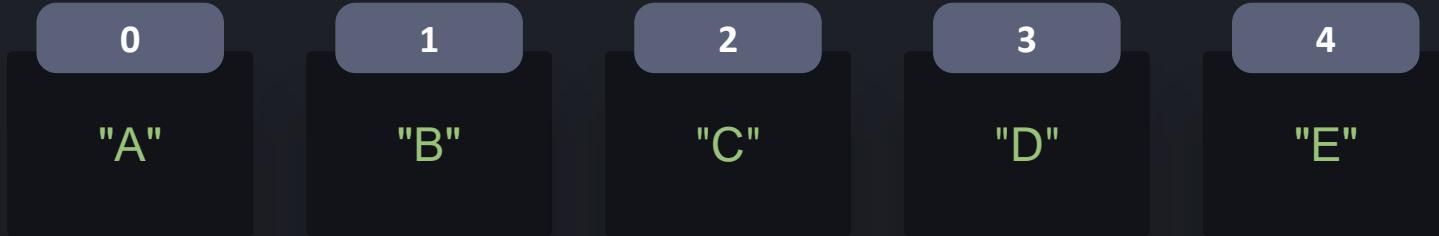
*element **in** list*

*element **not in** list*



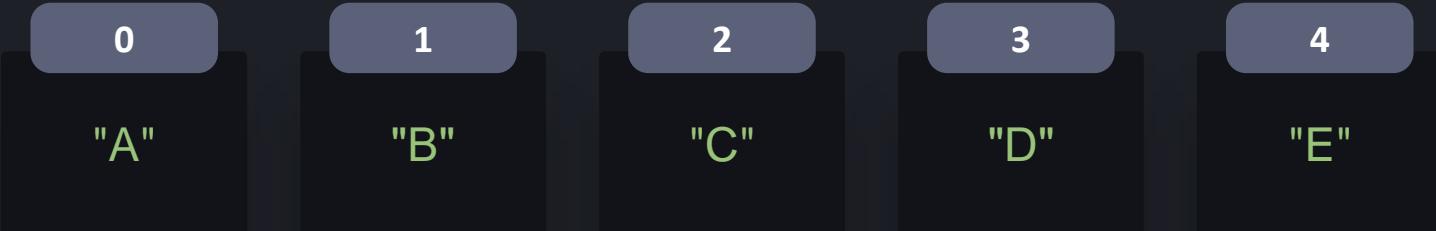
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> print("B" in letters)
```



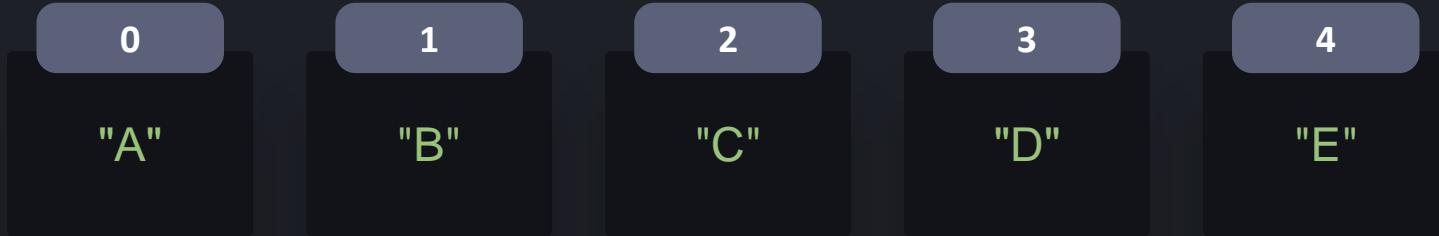
python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> print("B" in letters)
True
```



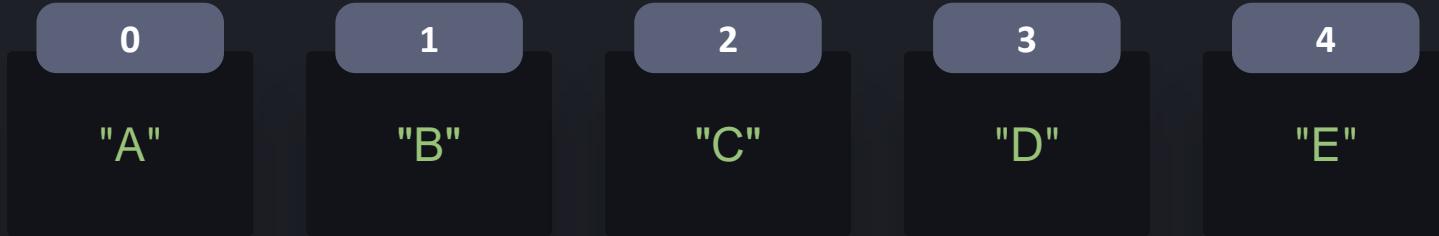
A screenshot of a terminal window titled "python3". The window has three colored circular icons in the top-left corner: red, yellow, and green. The terminal displays the following Python session:

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> print("B" in letters)
True
>>> print("Z" in letters)
False
```



python3

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> print("B" not in letters)
False
>>> print("Z" not in letters)
True
```



A screenshot of a terminal window titled "python3". The window has three colored circular icons in the top-left corner: red, yellow, and green. The terminal displays the following Python code and output:

```
>>> letters = ["A", "B", "C", "D", "E"]
>>> print("B" not in letters)
False
>>> print("Z" not in letters)
True
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Nested Lists - 2D (Matrix)

"Sam"

"Max"

"Joe"

"Anne"

"Sofie"

"Lisa"

"Tim"

"Sasha"

"Claire"

"Sara"

"Leo"

"Kim"

"Zoe"

"Guy"

"Anna"

"Eva"

"Sam"

"Max"

"Joe"

"Anne"

"Sofie"

"Lisa"

"Tim"

"Sasha"

"Claire"

"Sara"

"Leo"

"Kim"

"Zoe"

"Guy"

"Anna"

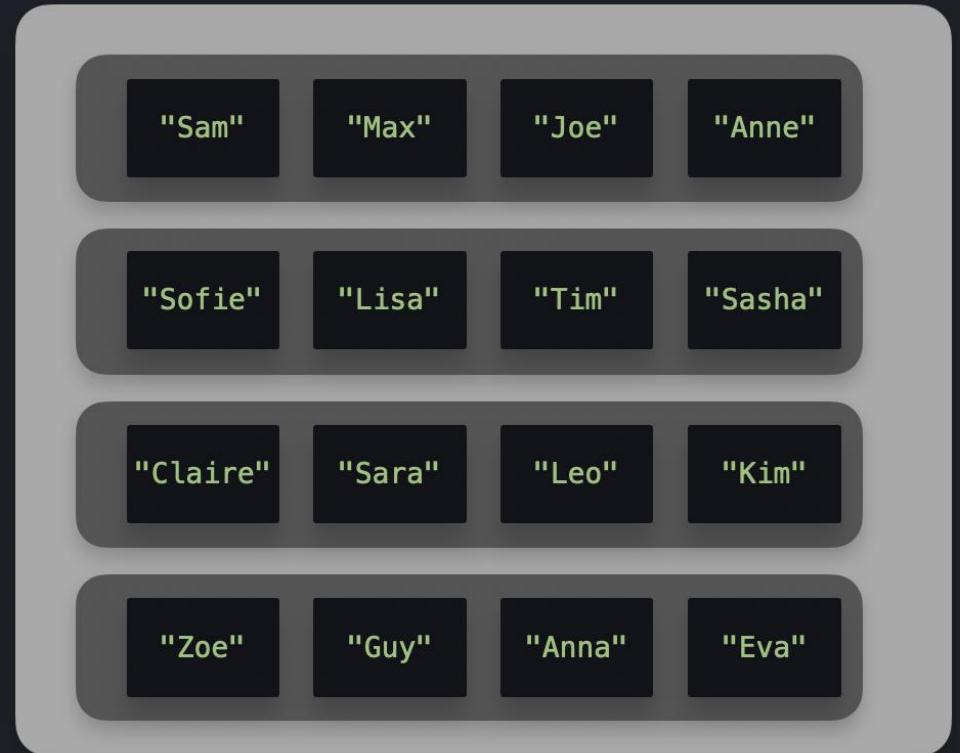
"Eva"



python3

>>>

```
classroom = [ ["Sam", "Max", "Joe", "Anne"],  
              ["Sofie", "Lisa", "Tim", "Sasha"],  
              ["Claire", "Sara", "Leo", "Kim"],  
              ["Zoe", "Guy", "Anna", "Eva"],  
          ]
```

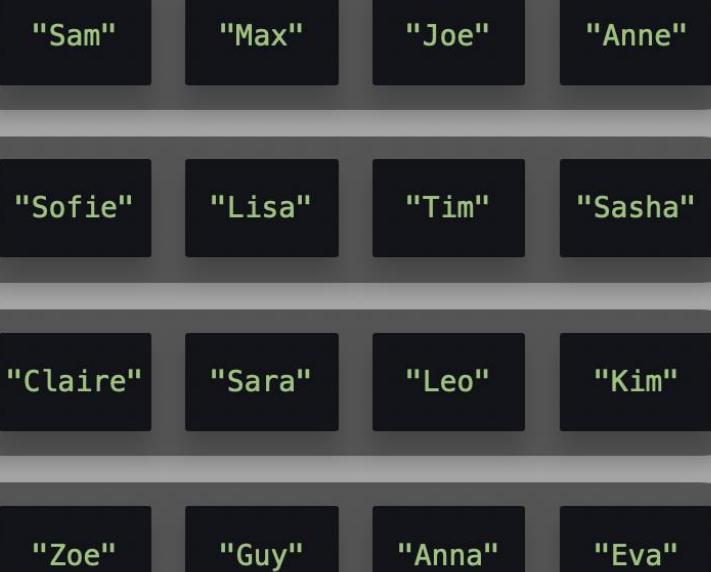




python3

>>>

```
classroom = [ ["Sam", "Max", "Joe", "Anne"],  
              ["Sofie", "Lisa", "Tim", "Sasha"],  
              ["Claire", "Sara", "Leo", "Kim"],  
              ["Zoe", "Guy", "Anna", "Eva"],  
          ]
```





python3

>>>

```
classroom = [ ["Sam", "Max", "Joe", "Anne"],  
              ["Sofie", "Lisa", "Tim", "Sasha"],  
              ["Claire", "Sara", "Leo", "Kim"],  
              ["Zoe", "Guy", "Anna", "Eva"],  
          ]
```

>>>

```
student = classroom[2]
```





python3

>>>

```
classroom = [
    ["Sam", "Max", "Joe", "Anne"],
    ["Sofie", "Lisa", "Tim", "Sasha"],
    ["Claire", "Sara", "Leo", "Kim"],
    ["Zoe", "Guy", "Anna", "Eva"],
]
```

>>>

```
student = classroom[2]
```





python3

>>>

```
classroom = [
    ["Sam", "Max", "Joe", "Anne"],
    ["Sofie", "Lisa", "Tim", "Sasha"],
    ["Claire", "Sara", "Leo", "Kim"],
    ["Zoe", "Guy", "Anna", "Eva"],
]
```

```
>>> student = classroom[2][1]
```





python3

>>>

```
classroom = [
    ["Sam", "Max", "Joe", "Anne"],
    ["Sofie", "Lisa", "Tim", "Sasha"],
    ["Claire", "Sara", "Leo", "Kim"],
    ["Zoe", "Guy", "Anna", "Eva"],
]
```

```
>>> student = classroom[2][1]
>>> print(student)
```

"Sam" "Max" "Joe" "Anne"

"Sofie" "Lisa" "Tim" "Sasha"

"Claire" "Sara" "Leo" "Kim"

"Zoe" "Guy" "Anna" "Eva"



python3

>>>

```
classroom = [
    ["Sam", "Max", "Joe", "Anne"],
    ["Sofie", "Lisa", "Tim", "Sasha"],
    ["Claire", "Sara", "Leo", "Kim"],
    ["Zoe", "Guy", "Anna", "Eva"],
]
```

```
>>> student = classroom[2][1]
```

```
>>> print(student)
```

"Sara"





KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Nested Lists - 3D (Cube)

"Sam"

"Max"

"Joe"

"Anne"

"Sofie"

"Lisa"

"Tim"

"Sasha"

"Claire"

"Sara"

"Leo"

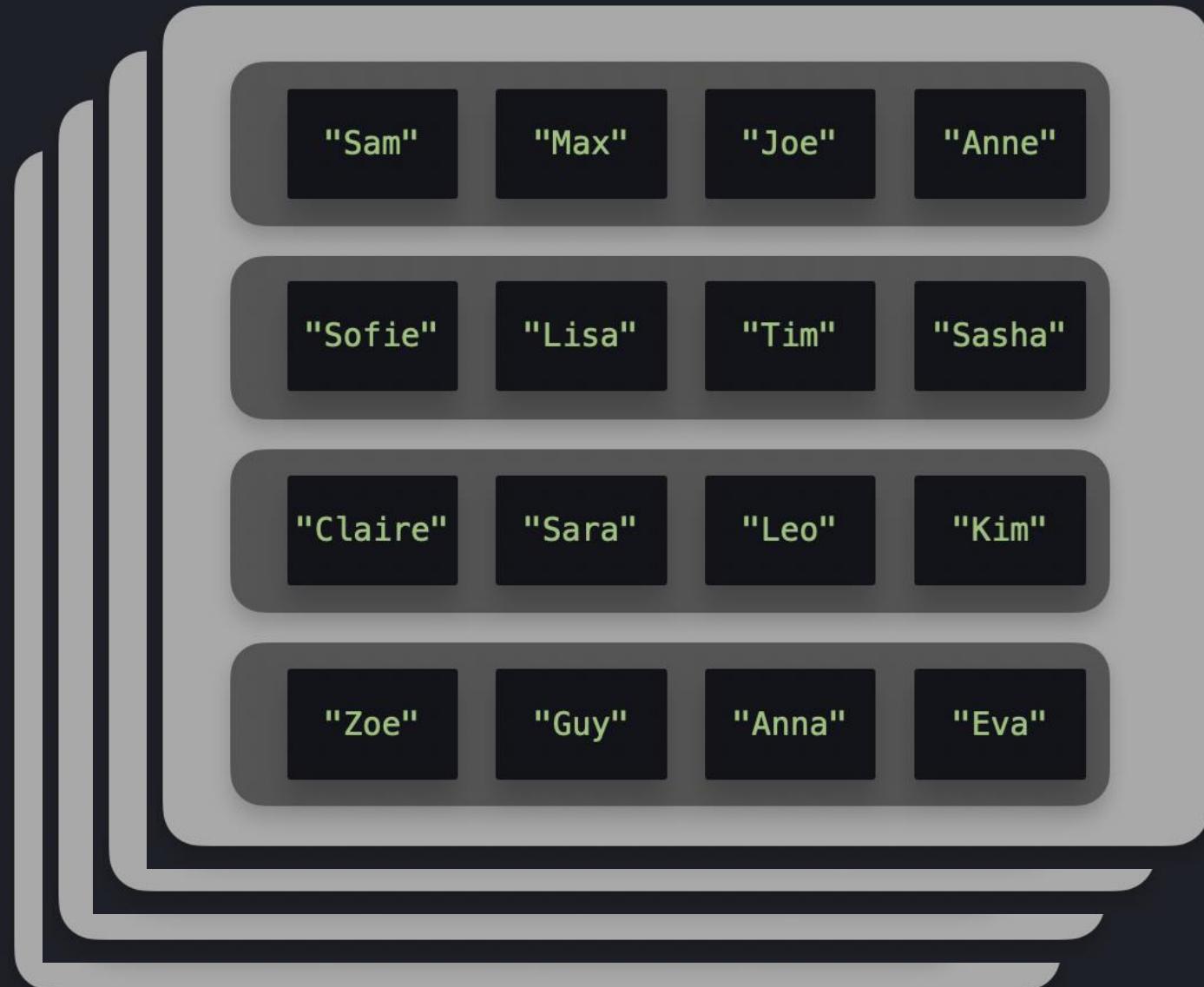
"Kim"

"Zoe"

"Guy"

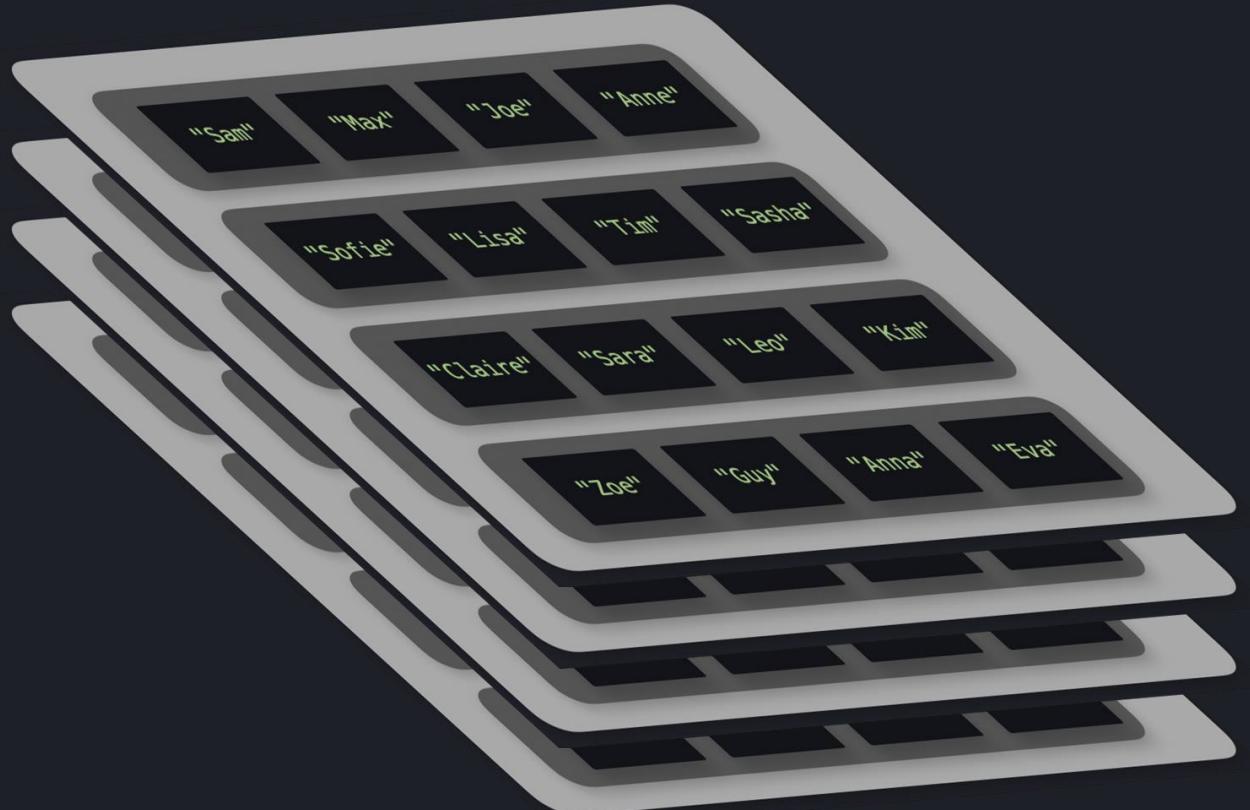
"Anna"

"Eva"



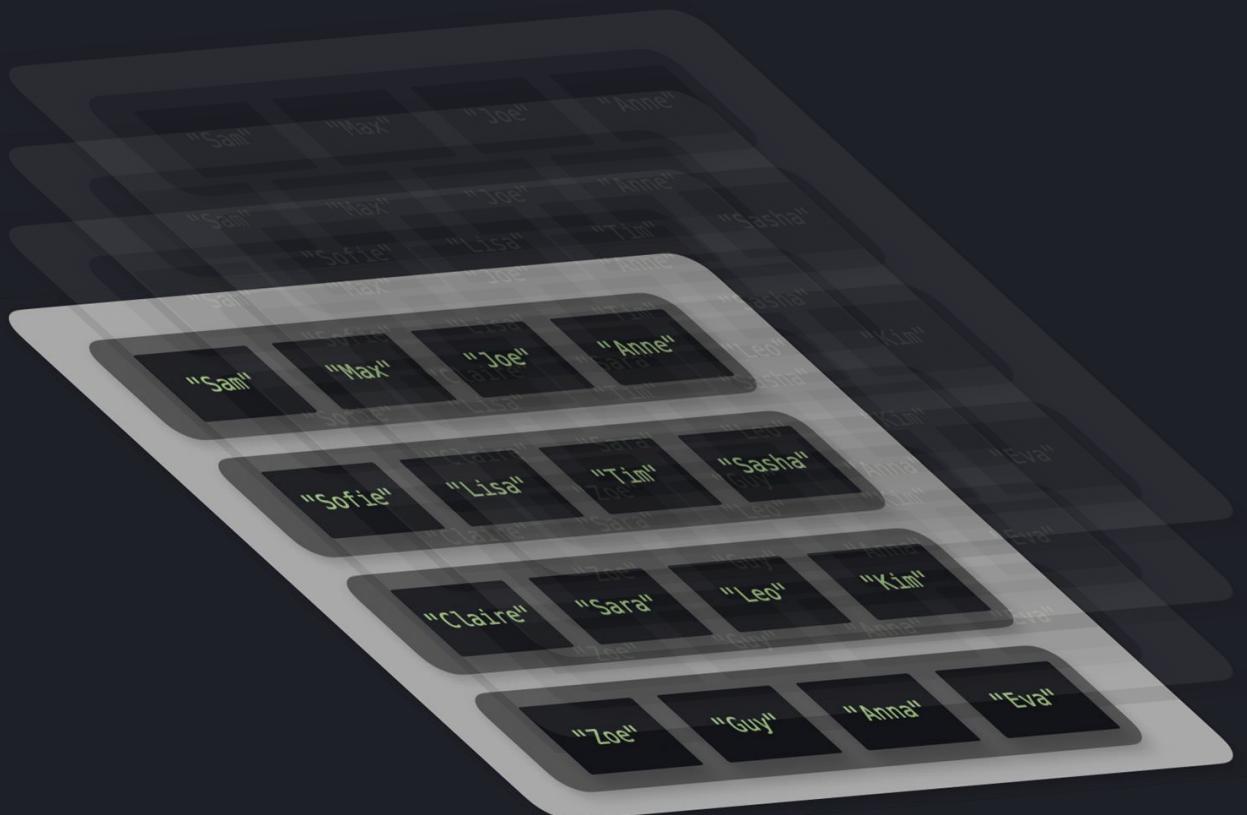


```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Guy", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ]
]
```





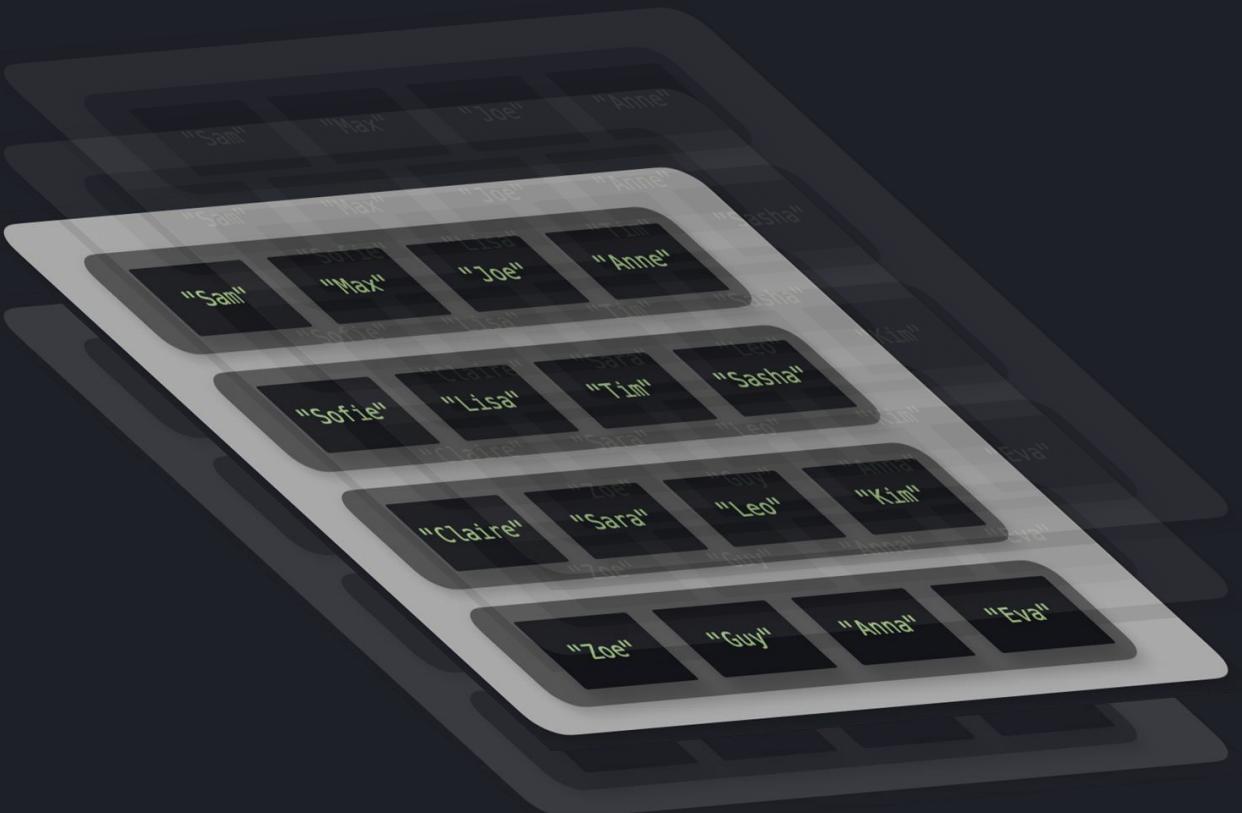
```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Guy", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ],
]
```





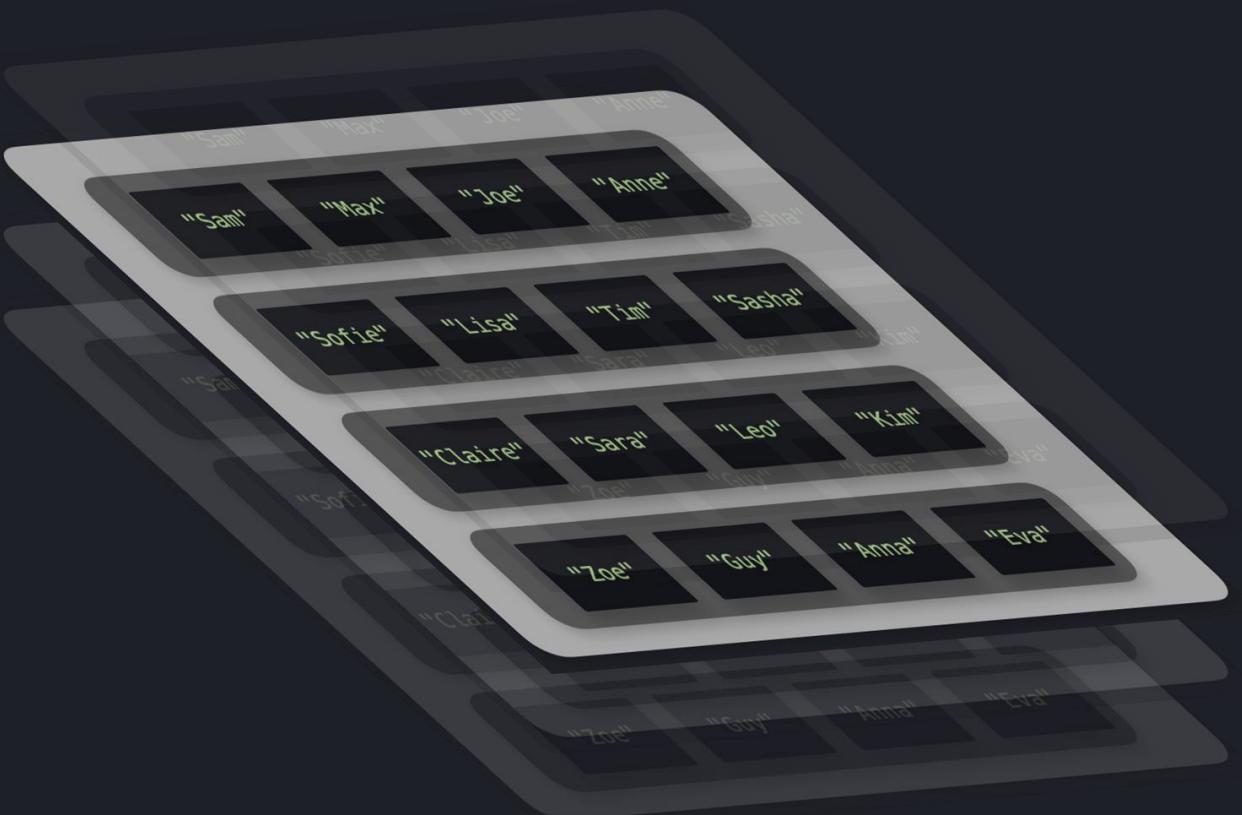
python3

```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Guy", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ]
]
```



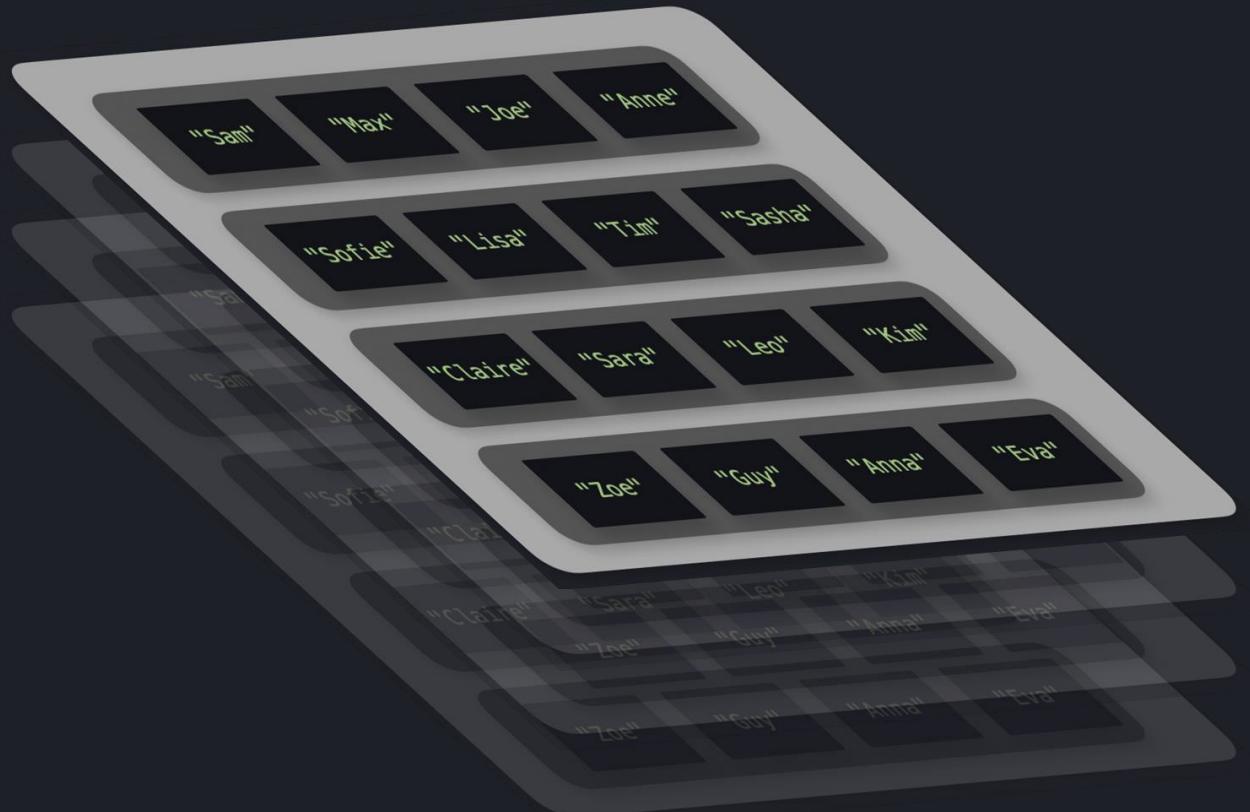


```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Guy", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ]
]
```





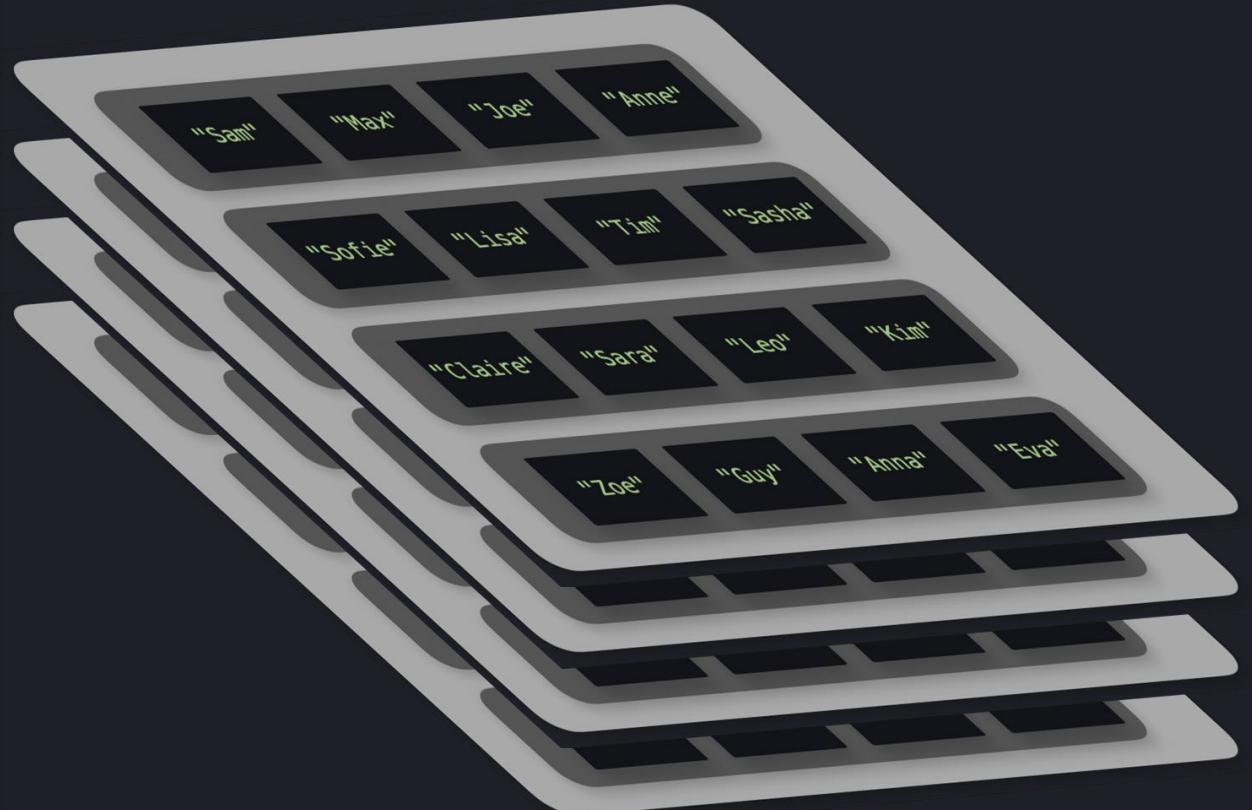
```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Guy", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ],
]
```





python3

```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Guy", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ],
]
```

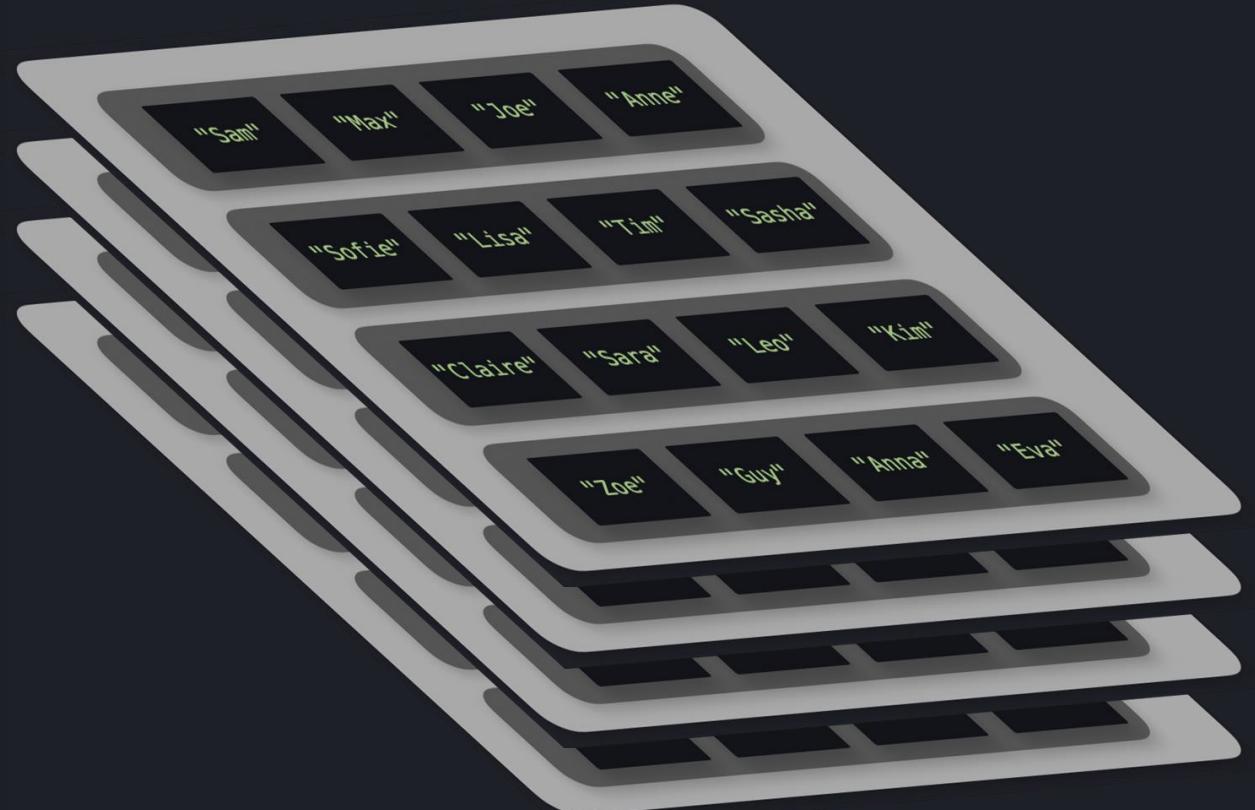




python3

```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Guy", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ],
]
]
```

Check out our full course on Python Entry-Level Programmer Certification (PCE P): <https://kode.wiki/3NnDTIU>

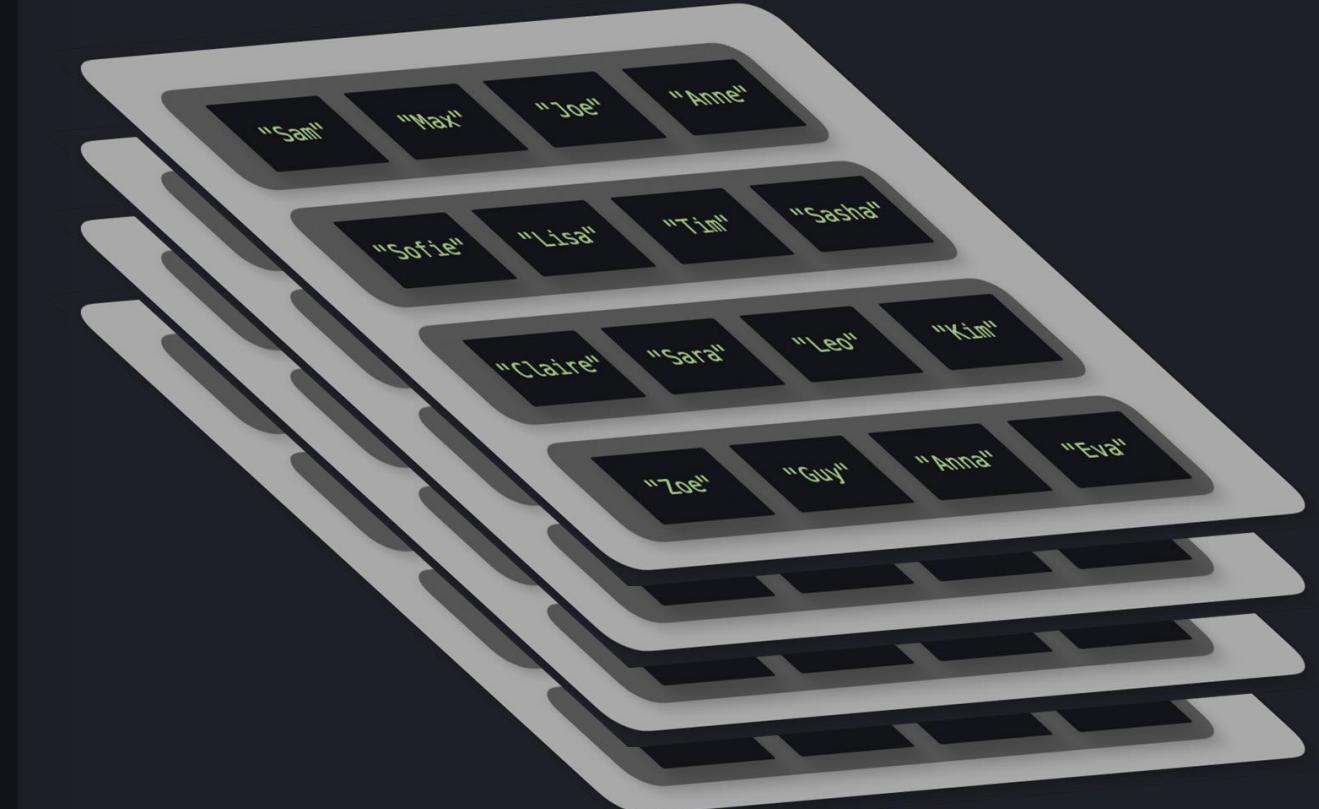




python3

```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Guy", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ],
]
]
```

>>>

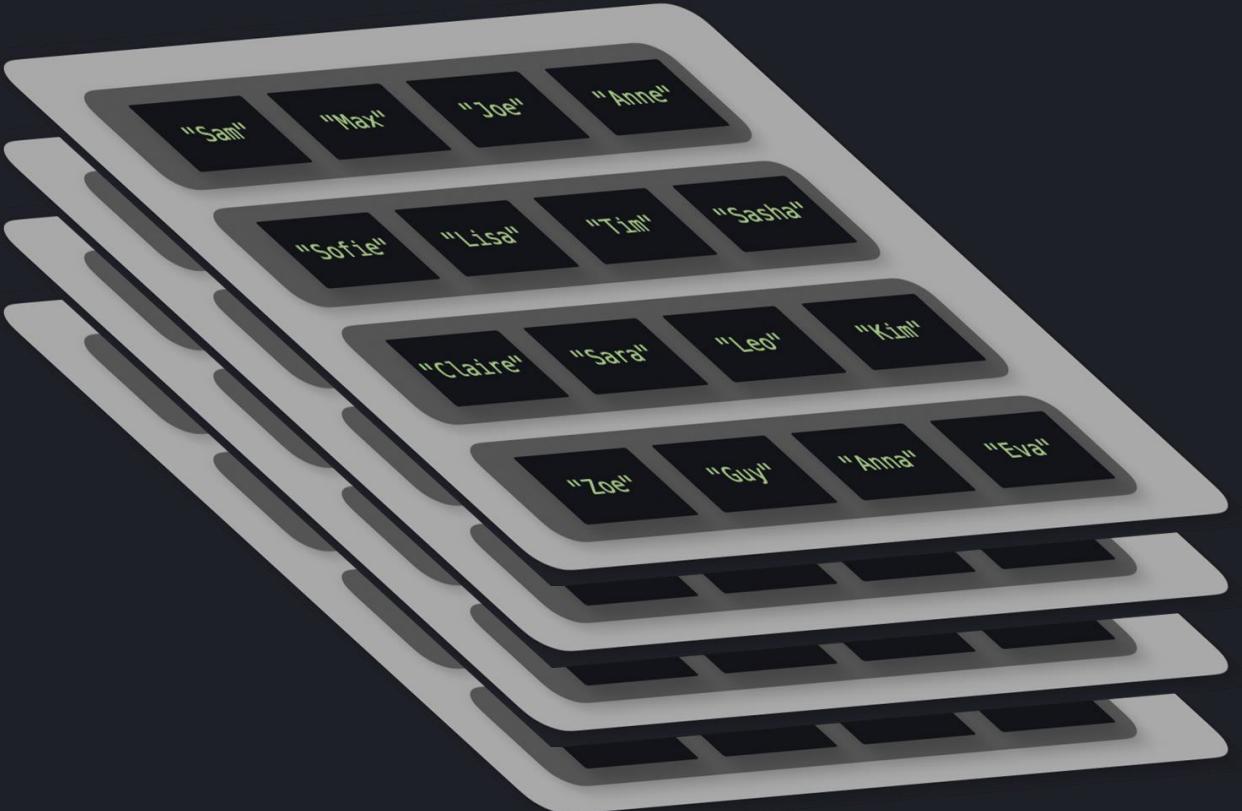


Get our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



python3

```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Guy", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ],
]
>>> student = school[1]
```

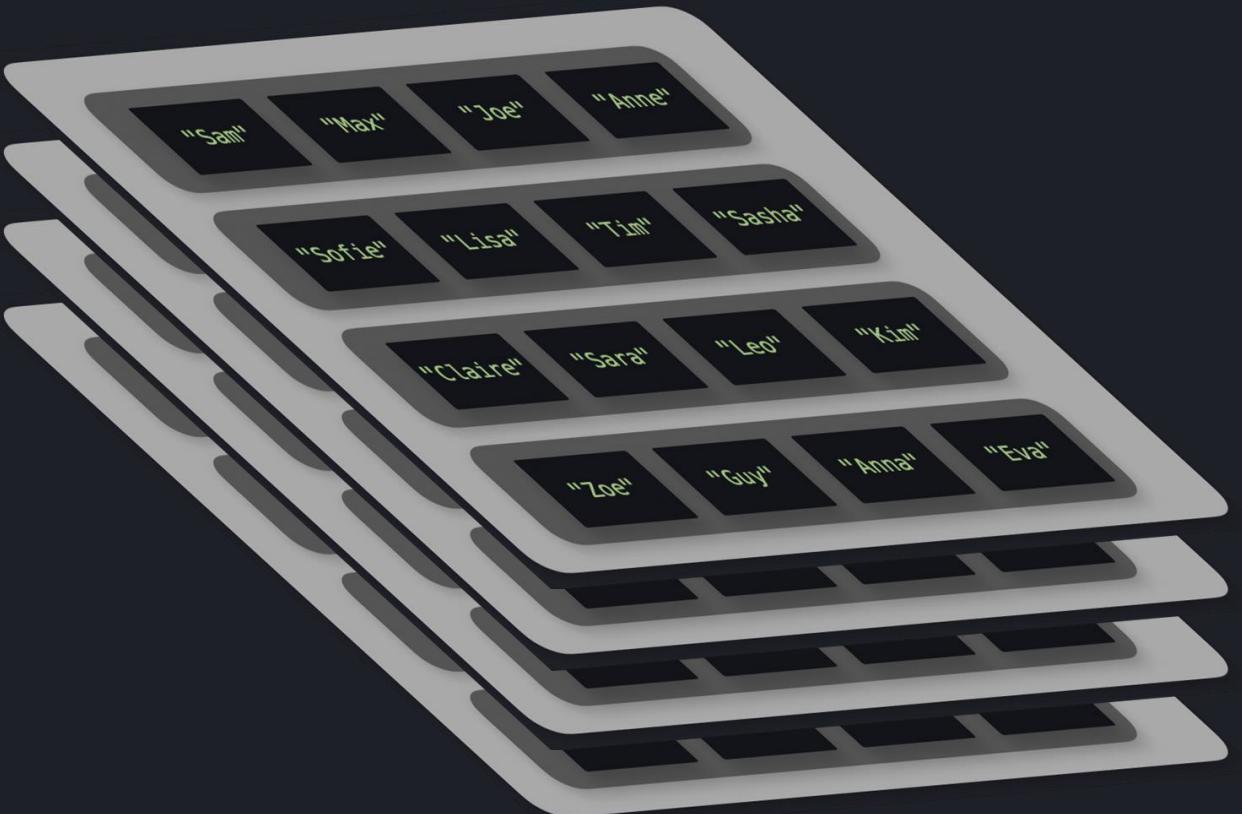




python3

```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Guy", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ],
]
]

>>> student = school[1]
```

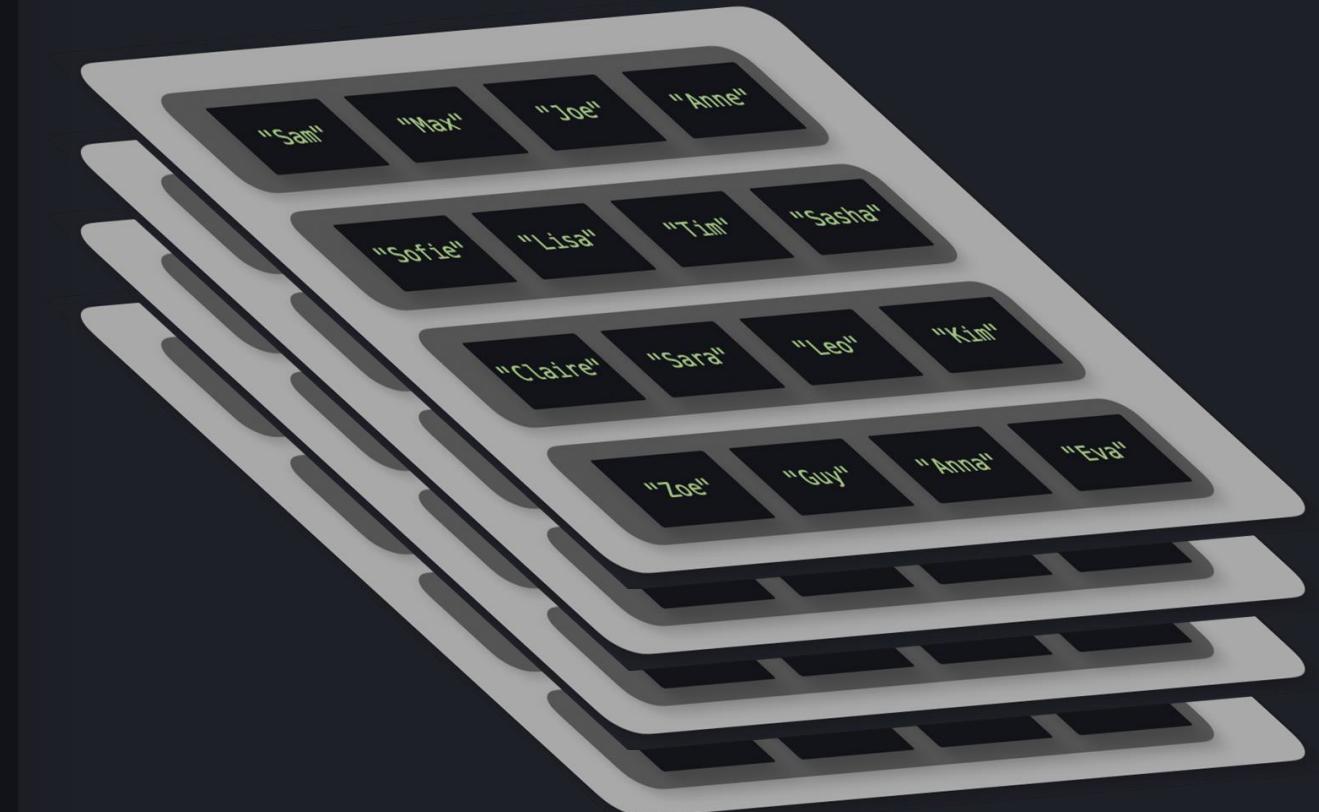




python3

```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Oscar", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ],
]
]

>>> student = school[1] [2]
```

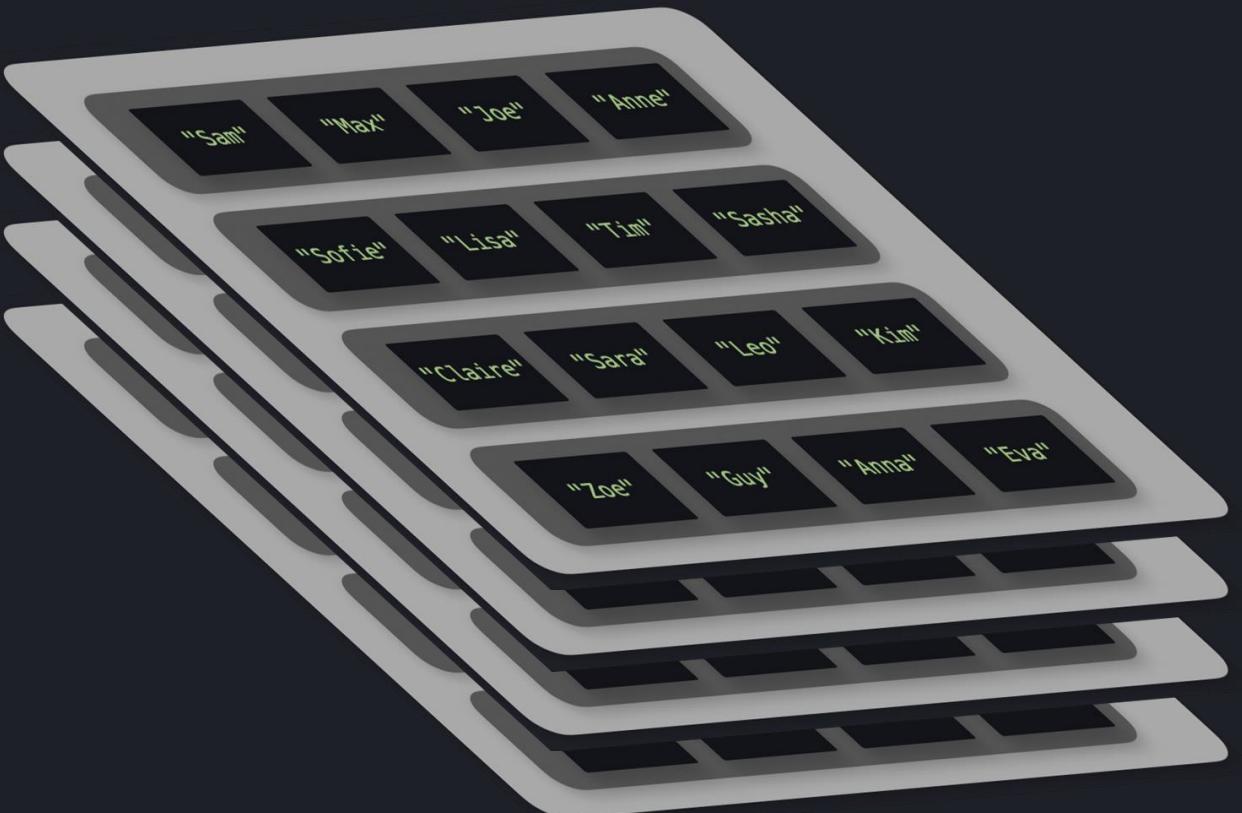




python3

```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Oscar", "Pete", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ],
]
]

>>> student = school[1] [2]
```

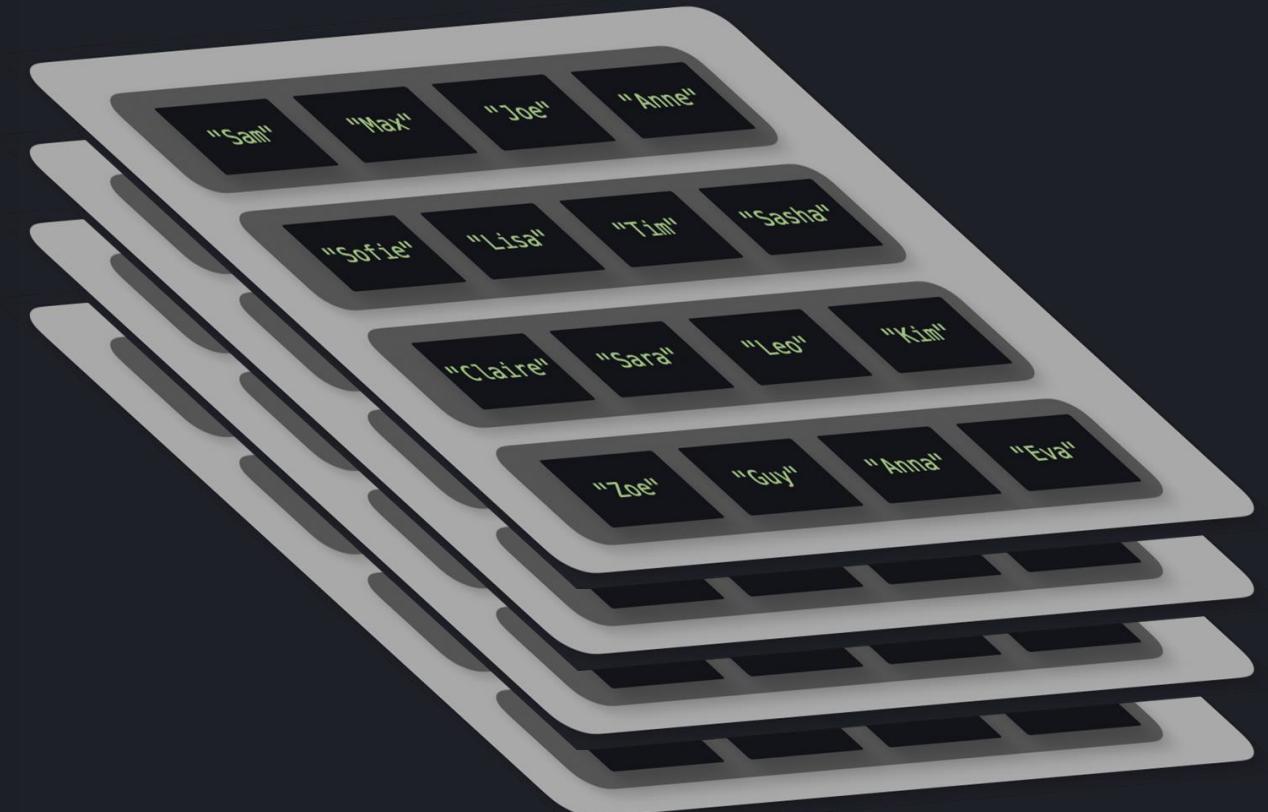




python3

```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Oscar", "Eva", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ],
]
]

>>> student = school[1] [2] [1]
```



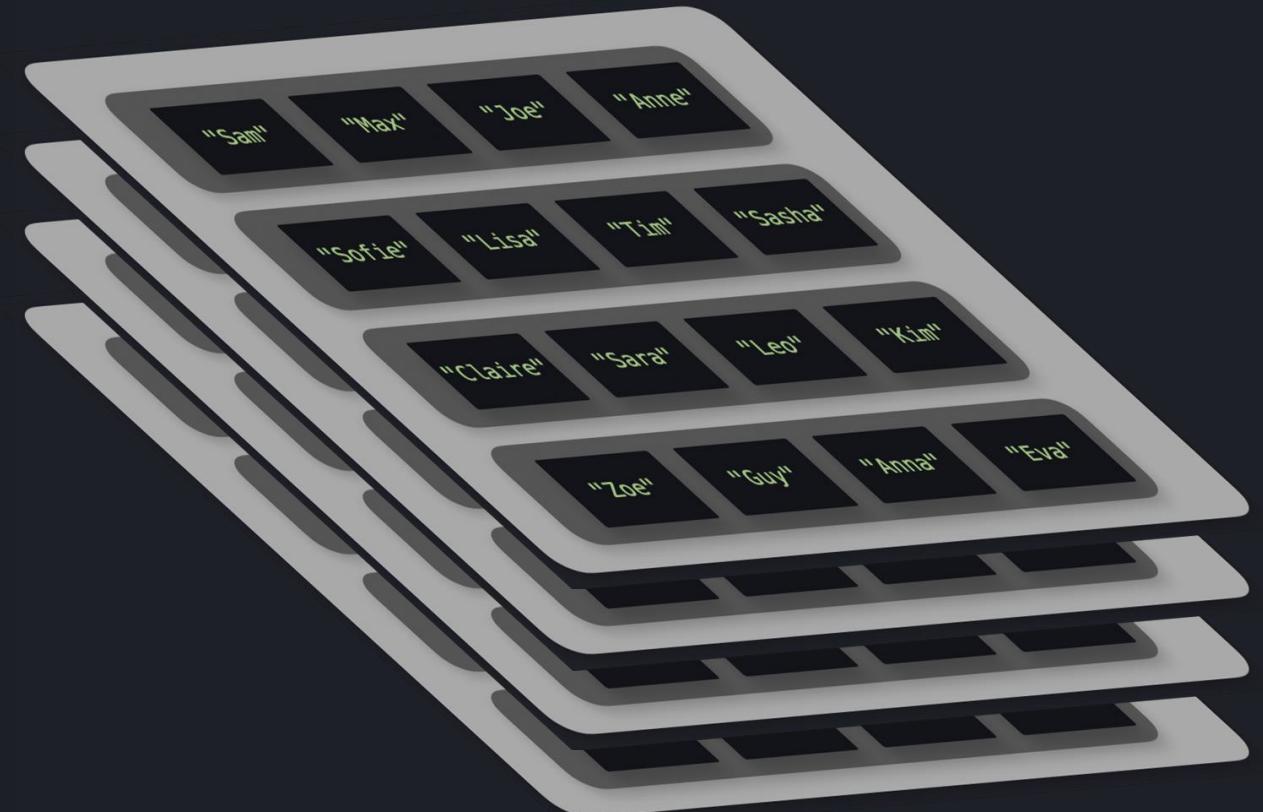
Check out our full course on Python Entry-Level Programmer Certification (PCE P): <https://kode.wiki/3NnDTIU>



python3

```
>>> school = [
    [
        ["Sara", "Kim", "Anne", "Eva"],
        ["Johan", "Collin", "Sam", "Alex"],
        ["Luke", "Sara", "Haley", "Jennifer"],
        ["Katy", "Mara", "Max", "Roy"],
    ],
    [
        ["Anne", "Leo", "Sasha", "Tim"],
        ["Claire", "Guy", "Pim", "Zoe"],
        ["Lisa", "Max", "Evan", "Chloe"],
        ["Brent", "Sam", "Sarah", "Anne"],
    ],
    [
        ["Maria", "Julian", "Chris", "Tom"],
        ["Zoe", "Anna", "Kim", "Leo"],
        ["Vera", "Pim", "Leo", "Guy"],
        ["Anne", "Sofie", "Max", "Joe"],
    ],
    [
        ["Sam", "Max", "Joe", "Anne"],
        ["Sofie", "Lisa", "Tim", "Sasha"],
        ["Claire", "Sara", "Leo", "Kim"],
        ["Zoe", "Guy", "Anna", "Eva"],
    ],
]
]

>>> student = school[1] [2] [1]
```

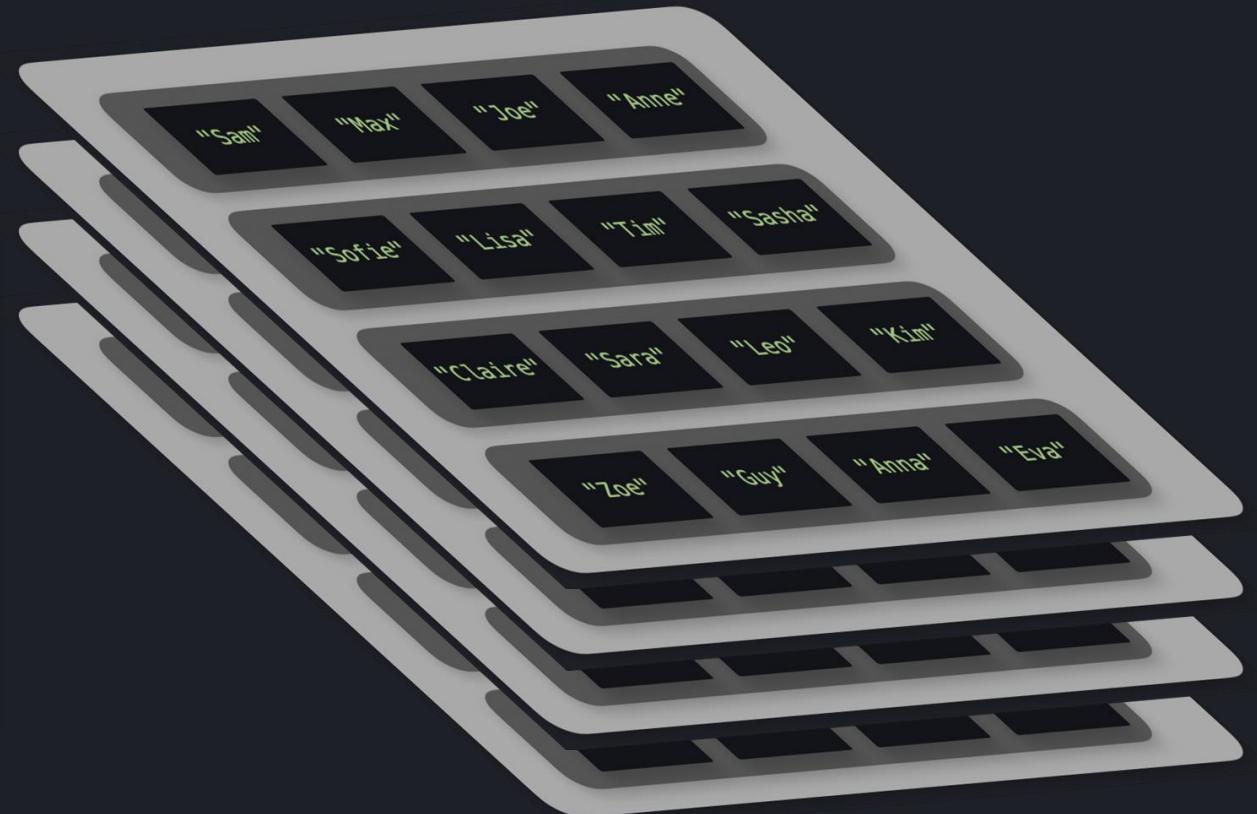


Check out our full course on Python Entry-Level Programmer Certification (PCE P): <https://kode.wiki/3NnDTIU>



python3

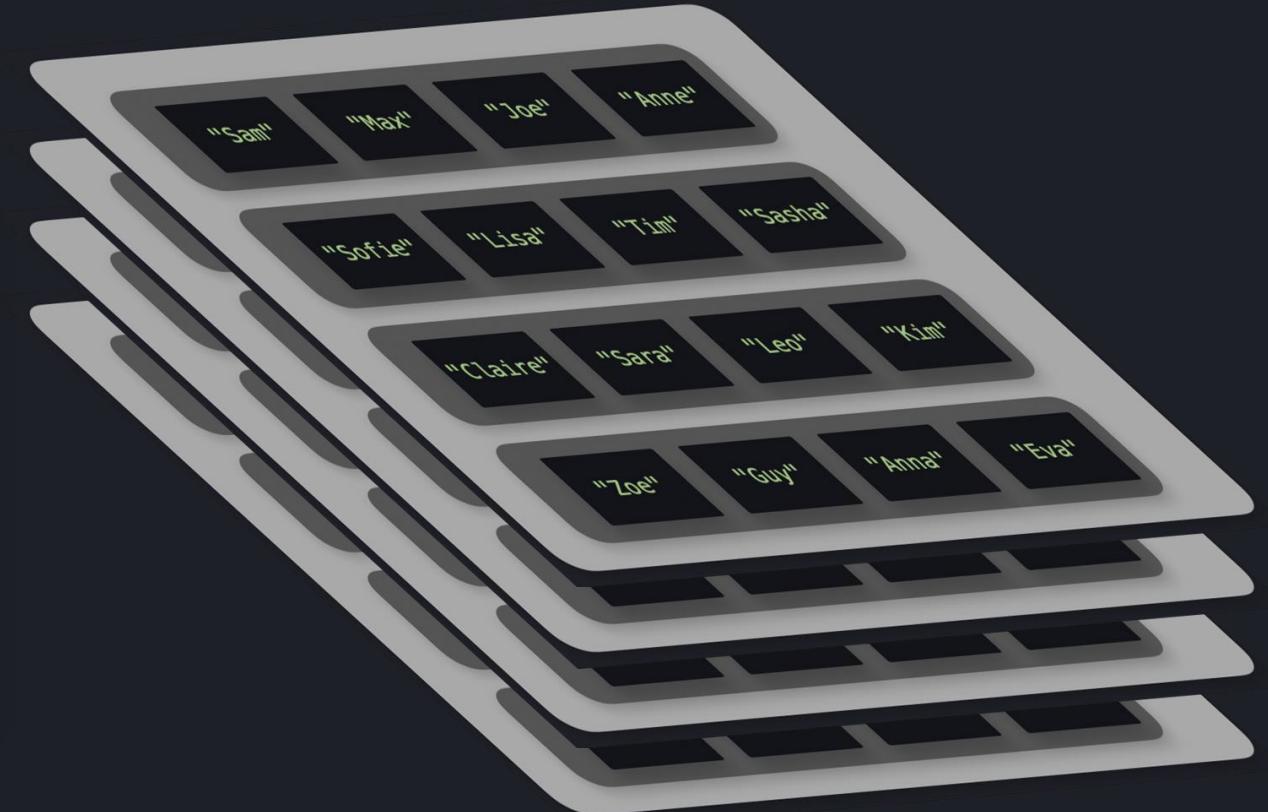
```
>>> student = school[1]      [2] [1]
>>> print(student)
```





python3

```
>>> student = school[1]      [2] [1]
>>> print(student)
"Max"
```





KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Functions

Functions

`print()`

`len()`

`input()`

Methods

`list.append()`

`list.insert()`



python3

```
>>> input1 = int(input("Enter a number: "))
```



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input2 = int(input("Enter a number: "))
```



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input2 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
```



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input2 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
```



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input2 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))
```



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input2 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))
```



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))
```



python3

```
>>> def input_number():
    return int(input("Enter a number: "))
```



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))
```

```
def input_number():
```

```
    return int(input("Enter a number: "))
```

>>>



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))
```

```
def input_number():
```

```
    return int(input("Enter a number: "))
```

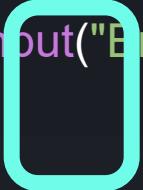
>>>



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))
```

```
def input_number():
```



```
    return int(input("Enter a number: "))
```

>>>



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))
def input_number():
    return int(input("Enter a number: "))
```





python3

```
>>> input1 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))
```

```
def input_number():
```

```
    return int(input("Enter a number: "))
```

```
>>>
```



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))
```

```
def input_number():
```

```
    return int(input("Enter a number: "))
```

```
>>>
```



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))
def input_number():
    return int(input("Enter a number: "))

>>>
```



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))

def input_number():
    return int(input("Enter a number: "))

>>>
```



python3

```
>>> def input_number():
    return int(input("Enter a number: "))
```



python3

```
>>> input1 = int(input("Enter a number: "))
>>> input2 = int(input("Enter a number: "))
>>> input3 = int(input("Enter a number: "))
>>> input4 = int(input("Enter a number: "))
>>> input5 = int(input("Enter a number: "))
```



python3

```
>>> def input_number():
        return int(input("Enter a number: "))

>>> input1 =    input_number()
>>> input2 =    input_number()
>>> input3 =    input_number()
>>> input4 =    input_number()
>>> input5 =    input_number()
```



python3

```
>>> def input_number():
        return int(input("Enter a number: "))

>>> input1 =    input_number()
>>> input2 =    input_number()
>>> input3 =    input_number()
>>> input4 =    input_number()
>>> input5 =    input_number()
```



python3

```
>>> def input_number():
        return int(input("Enter a number: "))

>>> input1 =    input_number()
>>> input2 =    input_number()
>>> input3 =    input_number()
>>> input4 =    input_number()
>>> input5 =    input_number()
>>> input1
```



python3

```
>>> def input_number():
        return int(input("Enter a number: "))

>>> input1 =    input_number()
>>> input2 =    input_number()
>>> input3 =    input_number()
>>> input4 =    input_number()
>>> input5 =    input_number()
>>> input1
Enter a number:    104
```



python3

```
>>> def input_number():
        return int(input("Enter a number: "))

>>> input1 =    input_number()
>>> input2 =    input_number()
>>> input3 =    input_number()
>>> input4 =    input_number()
>>> input5 =    input_number()
>>> input1
Enter a number:    104
>>> print(input1)
```



python3

```
>>> def input_number():
        return int(input("Enter a number: "))

>>> input1 =    input_number()
>>> input2 =    input_number()
>>> input3 =    input_number()
>>> input4 =    input_number()
>>> input5 =    input_number()
>>> input1
Enter a number:    104
>>> print(input1)
104
```



python3

```
>>> def input_number():
        return int(input("Enter a number: "))

>>> input1 =    input_number()
>>> input2 =    input_number()
>>> input3 =    input_number()
>>> input4 =    input_number()
>>> input5 =    input_number()
>>> input1
Enter a number:    104
>>> print(input1)
104
```



python3

```
>>> def input_number():
        return int(input("Enter a number: "))

>>> input1 =    input_number()
>>> input2 =    input_number()
>>> input3 =    input_number()
>>> input4 =    input_number()
>>> input5 =    input_number()
>>> input1
Enter a number:    104
>>> print(input1)
104
>>> input2
```



python3

```
>>> def input_number():
        return int(input("Enter a number: "))

>>> input1 =    input_number()
>>> input2 =    input_number()
>>> input3 =    input_number()
>>> input4 =    input_number()
>>> input5 =    input_number()
>>> input1
Enter a number:    104
>>> print(input1)
104
>>> input2
Enter a number:    34
```



python3

```
>>> def input_number():
        return int(input("Enter a number: "))

>>> input1 =    input_number()
>>> input2 =    input_number()
>>> input3 =    input_number()
>>> input4 =    input_number()
>>> input5 =    input_number()
>>> input1
Enter a number:    104
>>> print(input1)
104
>>> input2
Enter a number:    34
>>> print(input2)
```



python3

```
>>> def input_number():
        return int(input("Enter a number: "))

>>> input1 =    input_number()
>>> input2 =    input_number()
>>> input3 =    input_number()
>>> input4 =    input_number()
>>> input5 =    input_number()
>>> input1
Enter a number:    104
>>> print(input1)
104
>>> input2
Enter a number:    34
>>> print(input2)
34
```



python3

```
>>> def input_number():
    return int(input("Enter a number: "))

>>> input1 =      input_number()
>>> input2 =      input_number()
>>> input3 =      input_number()
>>> input4 =      input_number()
>>> input5 =      input_number()
>>> input1
Enter a number:    104
>>> print(input1)
104
>>> input2
Enter a number:    34
>>> print(input2)
34
```



python3

```
>>> input1 =      input_number()

def input_number():
    return int(input("Enter a number: "))
```



python3

```
>>> input1 =      input_number()

def input_number():
    return int(input("Enter a number: "))

NameError: name 'input_number' is not defined
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Function - Arguments



python3

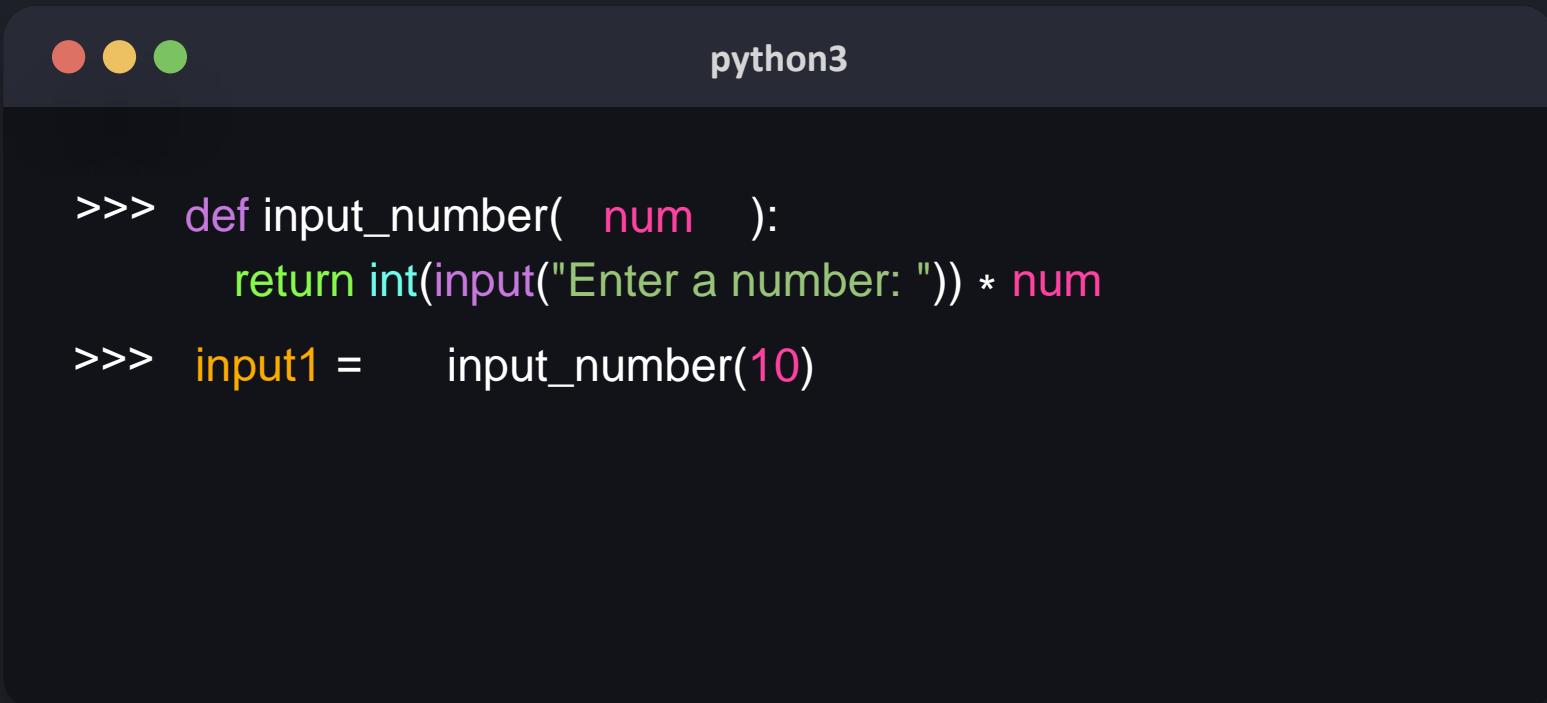
```
>>>
def input_number():
    return int(input("Enter a number: "))
```



python3

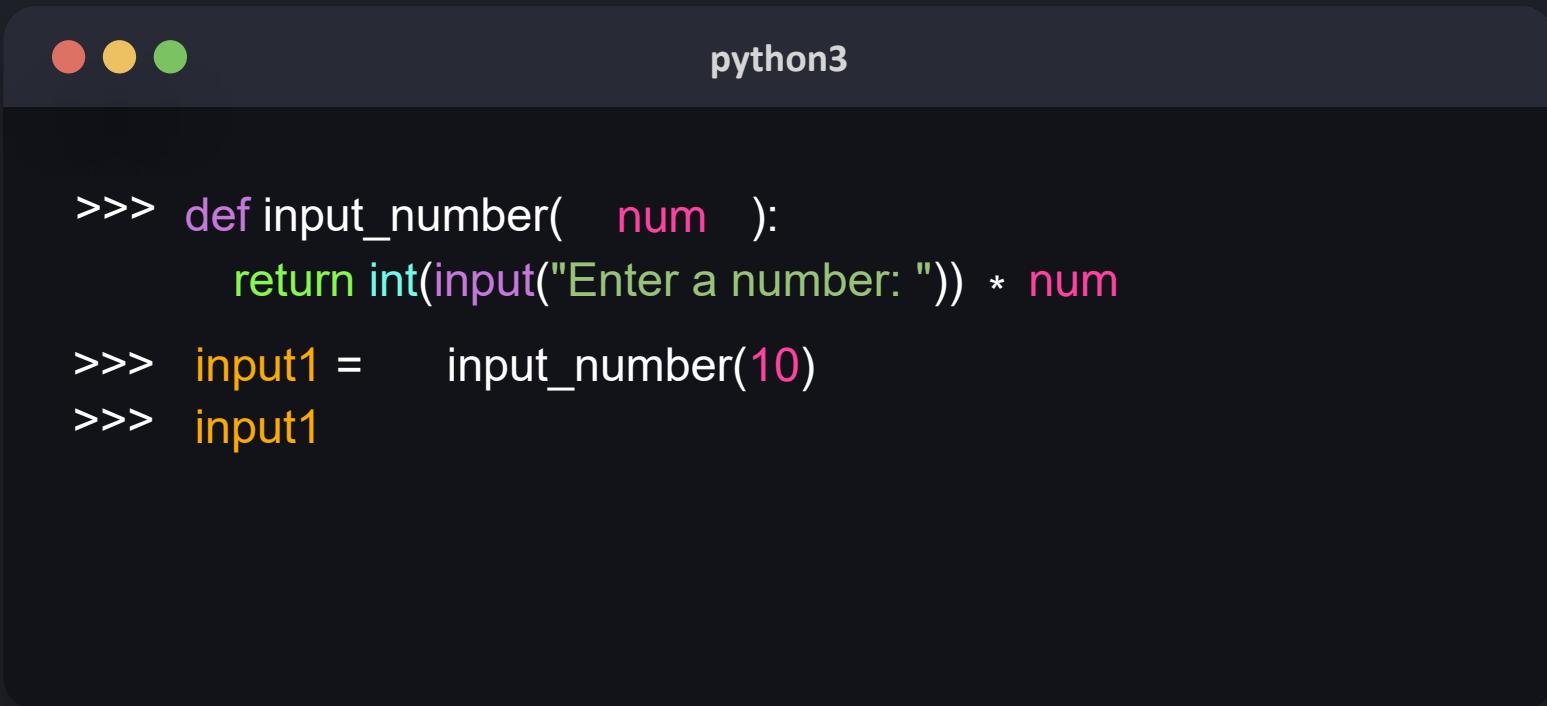
```
>>>
def input_number( num):
    return int(input("Enter a number: ")) * num
```

```
>>>  
def input_number(num ):  
    return int(input("Enter a number: ")) * num
```



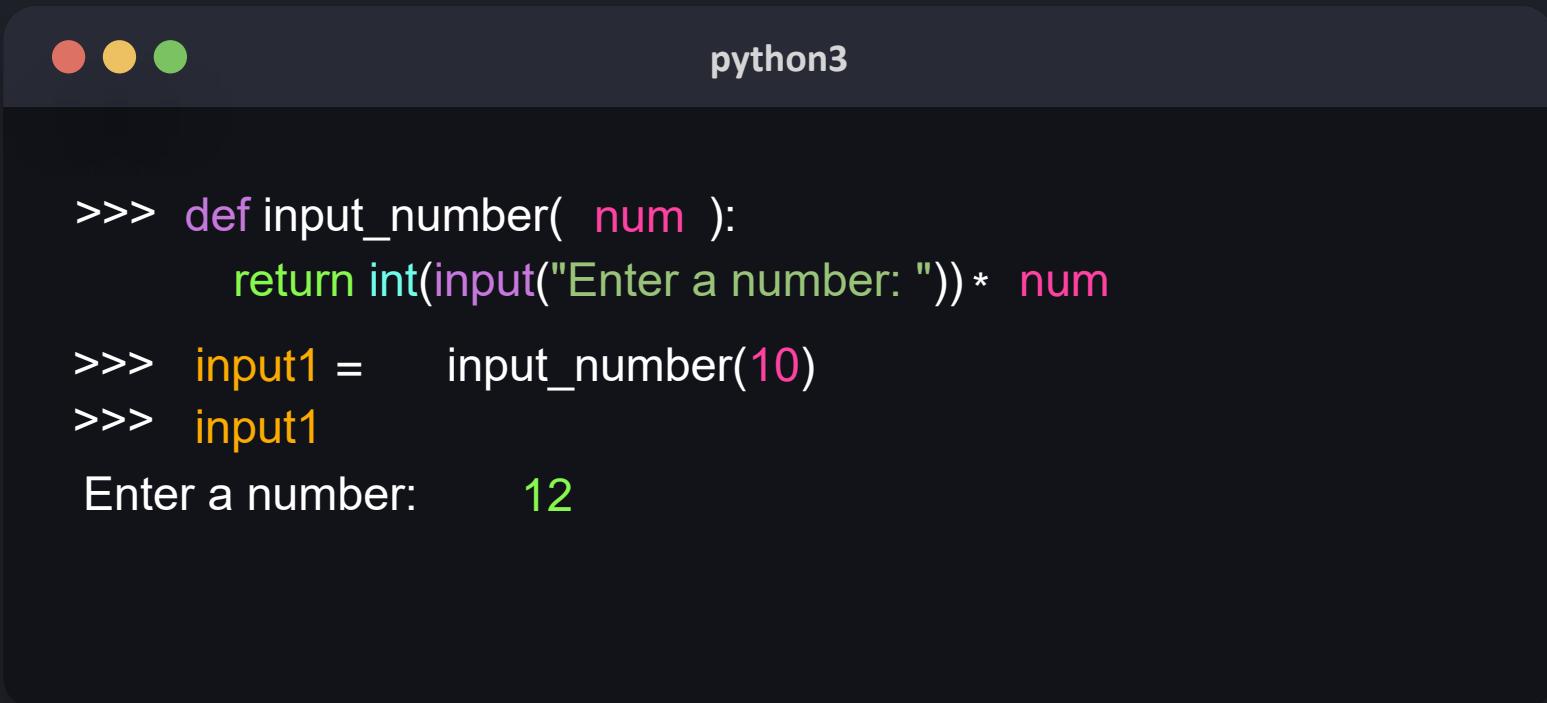
A screenshot of a macOS terminal window titled "python3". The window has a dark theme with red, yellow, and green window control buttons. The terminal displays the following Python code:

```
>>> def input_number( num ):  
    return int(input("Enter a number: ")) * num  
  
>>> input1 =     input_number(10)
```



A screenshot of a macOS terminal window titled "python3". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar is dark grey with the text "python3" in white. The main area of the terminal shows the following Python code:

```
>>> def input_number( num ):  
    return int(input("Enter a number: ")) * num  
  
>>> input1 = input_number(10)  
>>> input1
```

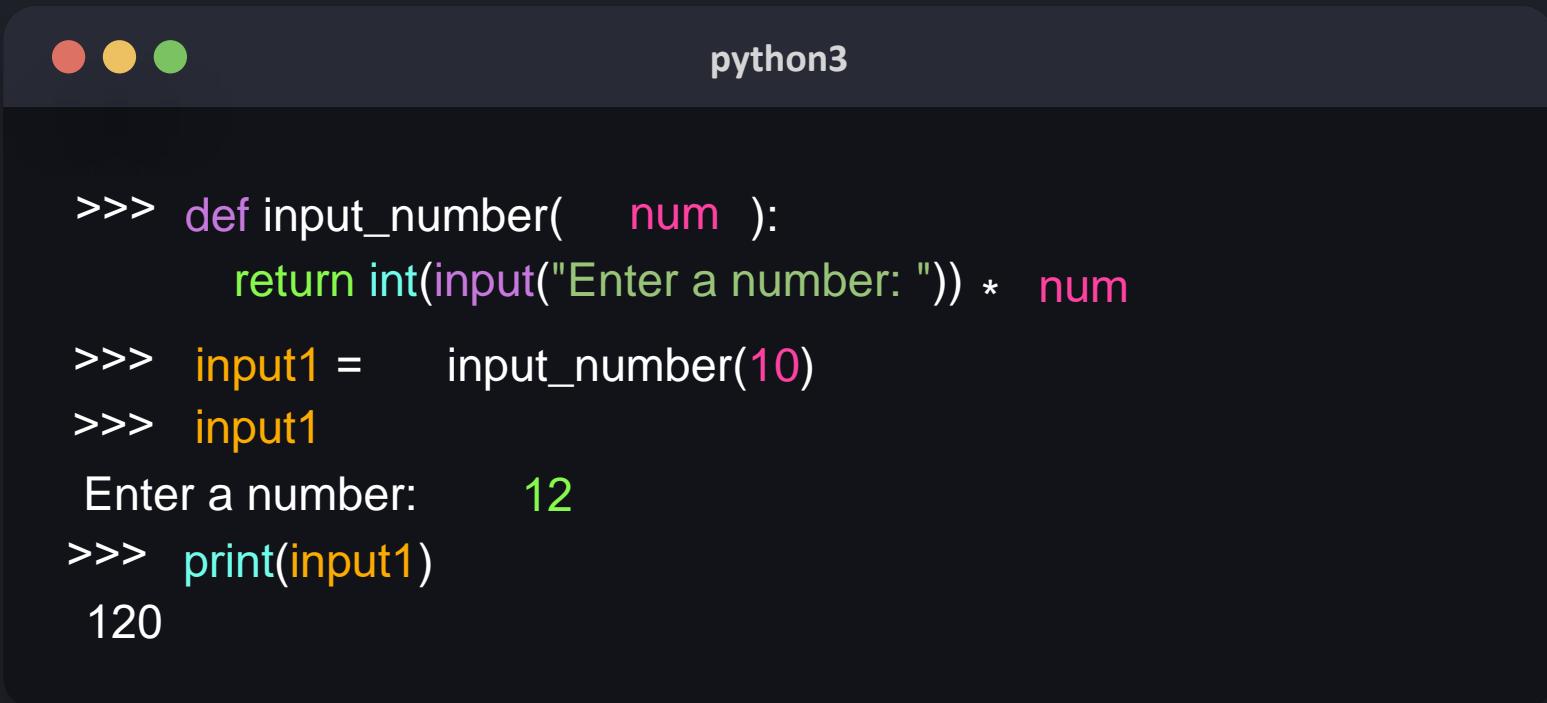


A screenshot of a macOS terminal window titled "python3". The window has a dark theme with red, yellow, and green window controls. The terminal displays the following Python code and its execution:

```
>>> def input_number( num ):
    return int(input("Enter a number: ")) * num
>>> input1 = input_number(10)
>>> input1
Enter a number: 12
```

 python3

```
>>> def input_number( num ):  
    return int(input("Enter a number: ")) * num  
  
>>> input1 = input_number(10)  
>>> input1  
Enter a number: 12  
>>> print(input1)
```



The image shows a screenshot of a macOS terminal window. The window title is "python3". The terminal contains the following Python code:

```
>>> def input_number( num ):
    return int(input("Enter a number: ")) * num

>>> input1 = input_number(10)
>>> input1
Enter a number: 12
>>> print(input1)
120
```



python3

```
>>> def input_number(num1, num2):  
    return int(input("Enter a number: ")) * num1 - num2
```



python3

```
>>> def input_number(num1, num2):  
    return int(input("Enter a number: ")) * num1 - num2  
>>> input1 =      input_number(10, 20)
```



python3

```
def input_number>>>(num1, num2):  
    return int(input("Enter a number: ")) * num1 - num2  
  
input1 =      input_number(    10,   20)
```



python3

```
def input_number(num1, num2):  
    return int(input("Enter a number")) * num1 - num2  
  
input1 = input_number(10, 20)
```



python3

```
def input_number(num1, num2):  
    return int(input("Enter a number: ")) * num1 - num2  
  
input1 =    input_number(    num2 = 10, num1 = 20)
```



python3

```
def input_number(num1, num2):  
    return int(input("Enter a number:")) * num1 - num2  
  
input1 =     input_number(      num2 = 10, num1 = 20)
```



python3

```
def input_number(num1, num2):  
    return int(input("Enter a number: ")) * num1 - num2
```

```
input1 =      input_number(10, num1 = 20)
```



python3

```
>>> def input_number(num1, num2):  
    return int(input("Enter a number: ")) * num1 - num2  
>>> input1 =     input_number(10, num1 = 20)
```

TypeError: input_number() got multiple values
for argument 'num1'



python3

```
>>> def input_number( num ):
    return int(input("Enter a number: ")) * num
```



python3

```
>>> def input_number(    num  = 10 ):  
     return int(input("Enter a number: ")) * num
```



python3

```
>>> def input_number( num = 10 ):
    return int(input("Enter a number: ")) * num
>>> input_number()
```



python3

```
>>> def input_number( num = 10 ):
    return int(input("Enter a number: ")) * num
>>> input_number()
Enter a number: 12
```



python3

```
>>> def input_number( num = 10 ):
    return int(input("Enter a number: ")) * num
>>> input_number()
Enter a number: 12
120
```



python3

```
>>> def input_number(    num  = 10 ):
        return int(input("Enter a number: ")) *  num
>>> input_number()
Enter a number:    12
120
```



python3

```
>>> def input_number(    num  = 10 ):
        return int(input("Enter a number: ")) * num
>>> input_number(5)
```



python3

```
>>> def input_number(    num  = 10 ):  
        return int(input("Enter a number: ")) *  num  
>>> input_number(5)  
Enter a number:    12
```



python3

```
>>> def input_number(    num  = 10 ):
        return int(input("Enter a number: ")) *  num
>>> input_number(5)
Enter a number:    12
60
```



python3

```
>>> def input_number( num = 10 ):
    return int(input("Enter a number: ")) * num
>>> input_number(5)
Enter a number: 12
60
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Function - Return



python3

```
>>> def input_number(    num  = 10 ):  
     return int(input("Enter a number: ")) *  num
```



python3

```
>>> def print_sum(      num1, num2  ):
    sum = num1 + num2
    print("The sum is: ", str(sum))
```



python3

```
>>> def print_sum(      num1, num2  ):
    sum = num1 + num2
    print("The sum is: ", str(sum))
>>> print_sum(10, 20)
```



python3

```
>>> def print_sum(      num1, num2 ):
    sum = num1 + num2
    print("The sum is: ", str(sum))
>>> print_sum(10, 20)
The sum is: 30
```



python3

```
>>> def print_sum(      num1, num2  ):
    sum = num1 + num2
    print("The sum is: ", str(sum))
>>> print_sum(10, 20)
The sum is: 30
```



python3

```
>>> def print_sum(      num1, num2 ):  
    sum = num1 + num2  
    return  
    print("The sum is: ", str(sum))
```



python3

```
>>> def print_sum(      num1, num2 ):  
    sum = num1 + num2  
    return  
    print("The sum is: ", str(sum))
```



python3

```
>>> def print_sum(      num1, num2 ):
    sum = num1 + num2

    if(sum == 0):
        return

    print("The sum is: ", str(sum))
```



python3

```
>>> def print_sum(      num1, num2  ):  
    sum = num1 + num2  
  
    if(sum == 0):  
        return  
  
    print("The sum is: ", str(sum))  
  
>>> print_sum(4, 2)
```



python3

```
>>> def print_sum(      num1, num2 ):  
    sum = num1 + num2  
  
    if(sum == 0):  
        return  
  
    print("The sum is: ", str(sum))
```

```
>>> print_sum(4, 2)  
The sum is: 6
```



python3

```
>>> def print_sum(      num1, num2 ):  
    sum = num1 + num2  
  
    if(sum == 0):  
        return  
  
    print("The sum is: ", str(sum))  
  
>>> print_sum(4, 2)  
The sum is: 6  
  
>>> print_sum(-1, 1)
```



python3

```
>>> def print_sum(      num1, num2 ):  
    sum = num1 + num2  
  
    if(sum == 0):  
        return  
  
    print("The sum is: ", str(sum))
```

```
>>> print_sum(4, 2)
```

```
The sum is: 6
```

```
>>> print_sum(-1, 1)
```

```
>>>
```



python3

```
>>> def print_sum(      num1, num2 ):  
    sum = num1 + num2  
  
    if(sum == 0):  
        return  
  
    print("The sum is: ", str(sum))  
  
>>> print_sum(4, 2)  
The sum is: 6  
  
>>> print_sum(-1, 1)  
>>>
```



python3

```
>>> def print_sum(      num1, num2 ):  
    sum = num1 + num2  
  
    if(sum == 0):  
        return  
  
    print("The sum is: ", str(sum))  
  
>>> print_sum(4, 2)  
The sum is: 6  
  
>>> print_sum(-1, 1)  
>>>
```



python3

```
>>> def is_even( num ):  
    if(num % 2 == 0):  
        return True
```



python3

```
>>> def is_even( num ):  
    if(num % 2 == 0):  
        return True  
  
>>> print(is_even(6))
```



python3

```
>>> def is_even( num ):  
    if(num % 2 == 0):  
        return True  
  
>>> print(is_even(6))  
True
```



python3

```
>>> def is_even( num ):  
    if(num % 2 == 0):  
        return True  
  
>>> print(is_even(6))  
True  
  
>>> print(is_even(7))
```



python3

```
>>> def is_even( num ):  
    if(num % 2 == 0):  
        return True  
  
>>> print(is_even(6))  
True  
  
>>> print(is_even(7))  
None
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Function - List as Argument



python3

```
>>> def multiply_values(list):
    multiplied_values = []

    for item in list:
        multiplied_values.append(item * 2)

    return multiplied_values
```



python3

```
>>> def multiply_values(list):
    multiplied_values = []

    for item in list:
        multiplied_values.append(item * 2)

    return multiplied_values
```



python3

```
>>> def multiply_values(list):
    multiplied_values = []

    for item in list:
        multiplied_values.append(item * 2)

    return multiplied_values
```



python3

```
>>> def multiply_values(list):
    multiplied_values = []
    for item in list:
        multiplied_values.append(item * 2)
    return multiplied_values
```



python3

```
>>> def multiply_values(list):
    multiplied_values = []

    for item in list:
        multiplied_values.append(item * 2)

    return multiplied_values
```



python3

```
>>> def multiply_values(list):
    multiplied_values = []

    for item in list:
        multiplied_values.append(item * 2)

    return multiplied_values
```



python3

```
>>> def multiply_values(list):
    multiplied_values = []

    for item in list:
        multiplied_values.append(item * 2)

    return multiplied_values

>>> print(multiply_values([1, 2, 3]))
```



python3

```
>>> def multiply_values(list):
    multiplied_values = []

    for item in list:
        multiplied_values.append(item * 2)

    return multiplied_values

>>> print(multiply_values([1, 2, 3]))
[2, 4, 6]
```



python3

```
>>> def multiply_values(list):
    multiplied_values = []

    for item in list:
        multiplied_values.append(item * 2)

    return multiplied_values

>>> print(multiply_values([1, 2, 3]))
[2, 4, 6]
>>> print(multiply_values([-4, -8, -10]))
```



python3

```
>>> def multiply_values(list):
    multiplied_values = []

    for item in list:
        multiplied_values.append(item * 2)

    return multiplied_values

>>> print(multiply_values([1, 2, 3]))
[2, 4, 6]
>>> print(multiply_values([-4, -8, -10]))
[-8, -16, -20]
```



python3

```
>>> def multiply_values(list):
    multiplied_values = []

    for item in list:
        multiplied_values.append(item * 2)

    return multiplied_values

>>> print(multiply_values([1, 2, 3]))
[2, 4, 6]
>>> print(multiply_values([-4, -8, -10]))
[-8, -16, -20]
>>> print(multiply_values(1))
```



python3

```
>>> def multiply_values(list):
    multiplied_values = []

    for item in list:
        multiplied_values.append(item * 2)

    return multiplied_values

>>> print(multiply_values([1, 2, 3]))
[2, 4, 6]
>>> print(multiply_values([-4, -8, -10]))
[-8, -16, -20]
>>> print(multiply_values(1))
```

TypeError: 'int' object is not iterable



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Scopes



python3

```
>>> def input_number():
    result = int(input("Enter a number: "))
    * 100
    return result
```



python3

```
>>> def input_number():
    result = int(input("Enter a number: "))
    * 100
    return result
```



python3

```
>>> def input_number():
    result = int(input("Enter a number: "))
    return result * 100
>>> print(result)
```



python3

```
>>> def input_number():
    result = int(input("Enter a number: "))
    return result
>>> print(result)
```

NameError: name 'result' is not defined



python3

```
>>> num = 100  
>>> def input_number():  
    result = int(input("Enter a number: "))  
    return result
```

* nu
 m



python3

```
>>> num = 100  
>>> def input_number():  
    num = 50  
    result = int(input("Enter a number: "))  
    return result
```

* nu
 m



python3

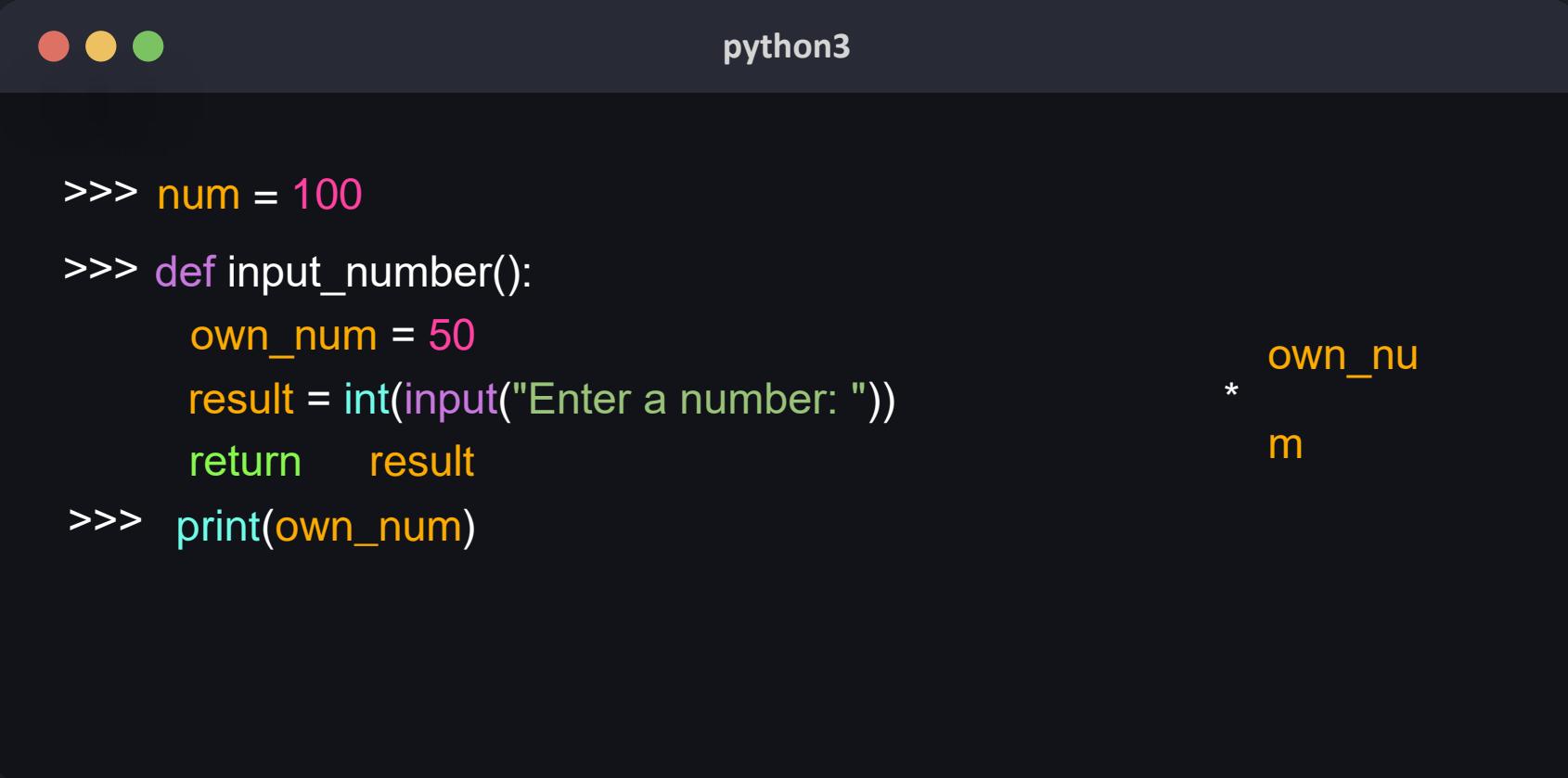
```
>>> num = 100  
>>> def input_number():  
    num = 50  
    result = int(input("Enter a number: "))  
    return result
```

* nu
 m



python3

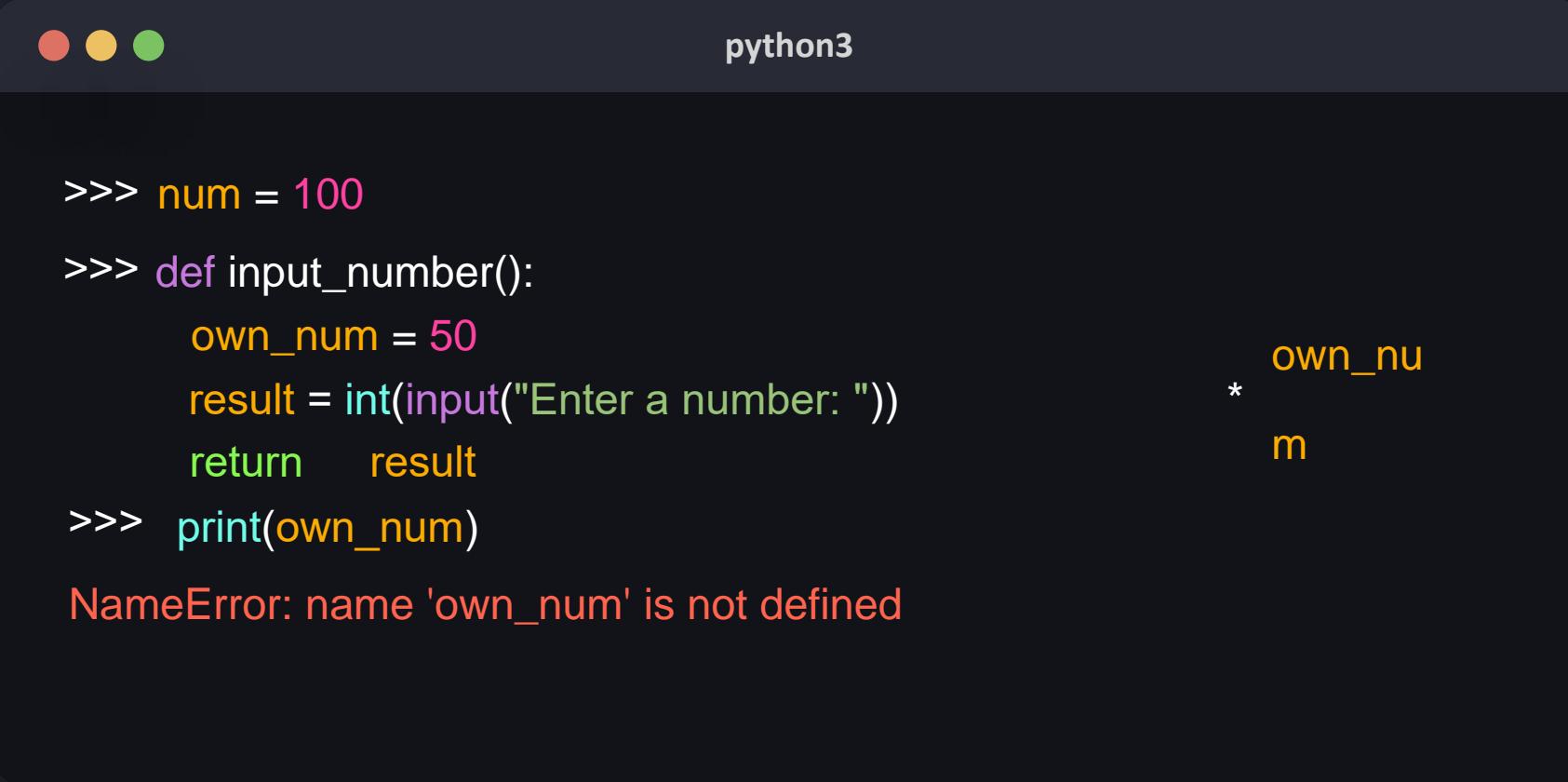
```
>>> num = 100
>>> def input_number():
    result = int(input("Enter a number: "))
    return    result
          nu
          *
          m
```



A screenshot of a macOS terminal window titled "python3". The window has three red, yellow, and green close buttons at the top left. The title bar is dark grey. The terminal itself is black with white text. It contains the following Python code:

```
>>> num = 100
>>> def input_number():
    own_num = 50
    result = int(input("Enter a number: "))
    return result
>>> print(own_num)
```

* own_nu
m

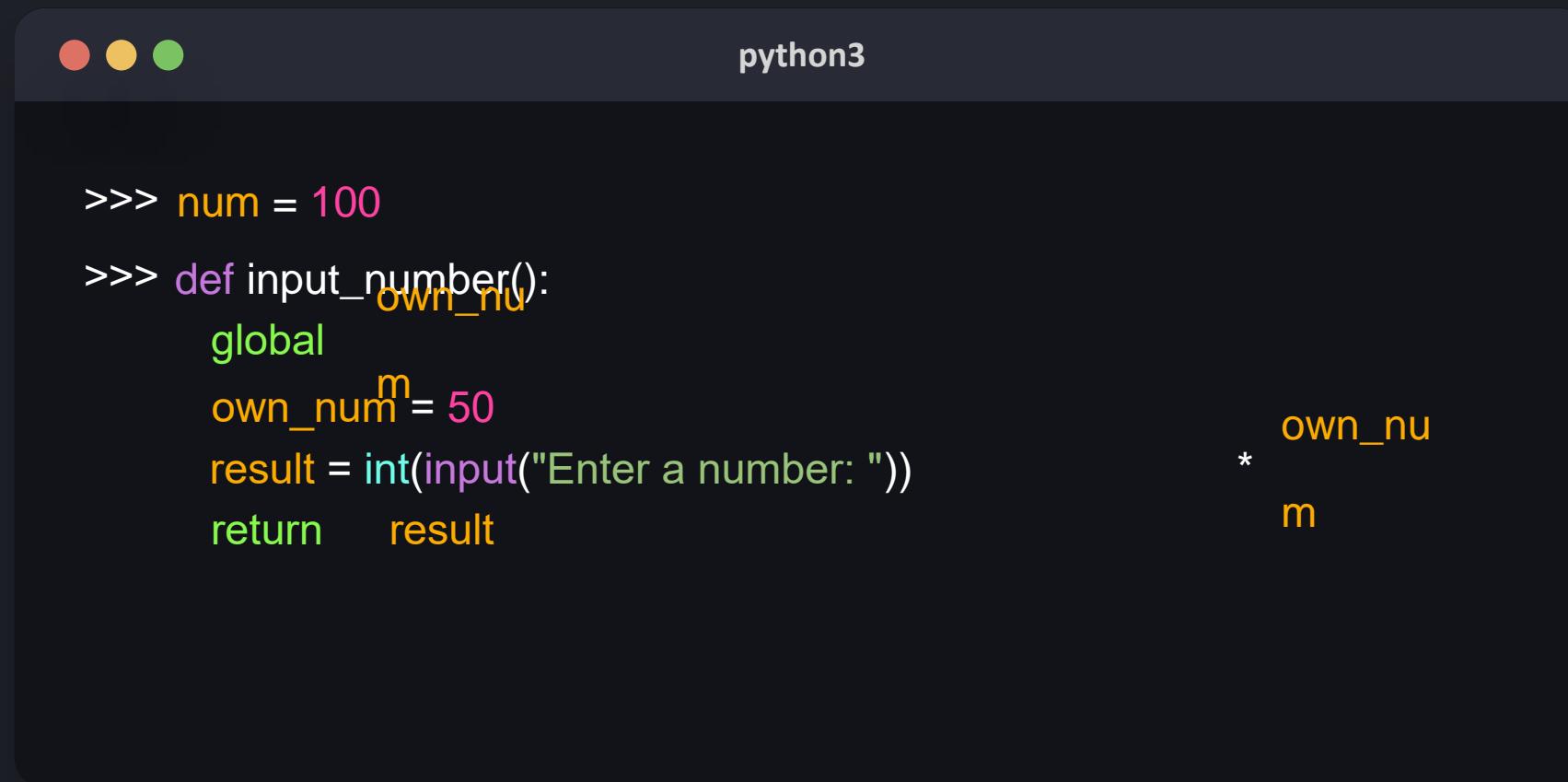


The image shows a screenshot of a macOS terminal window. The window title is "python3". Inside the terminal, the following Python code is being run:

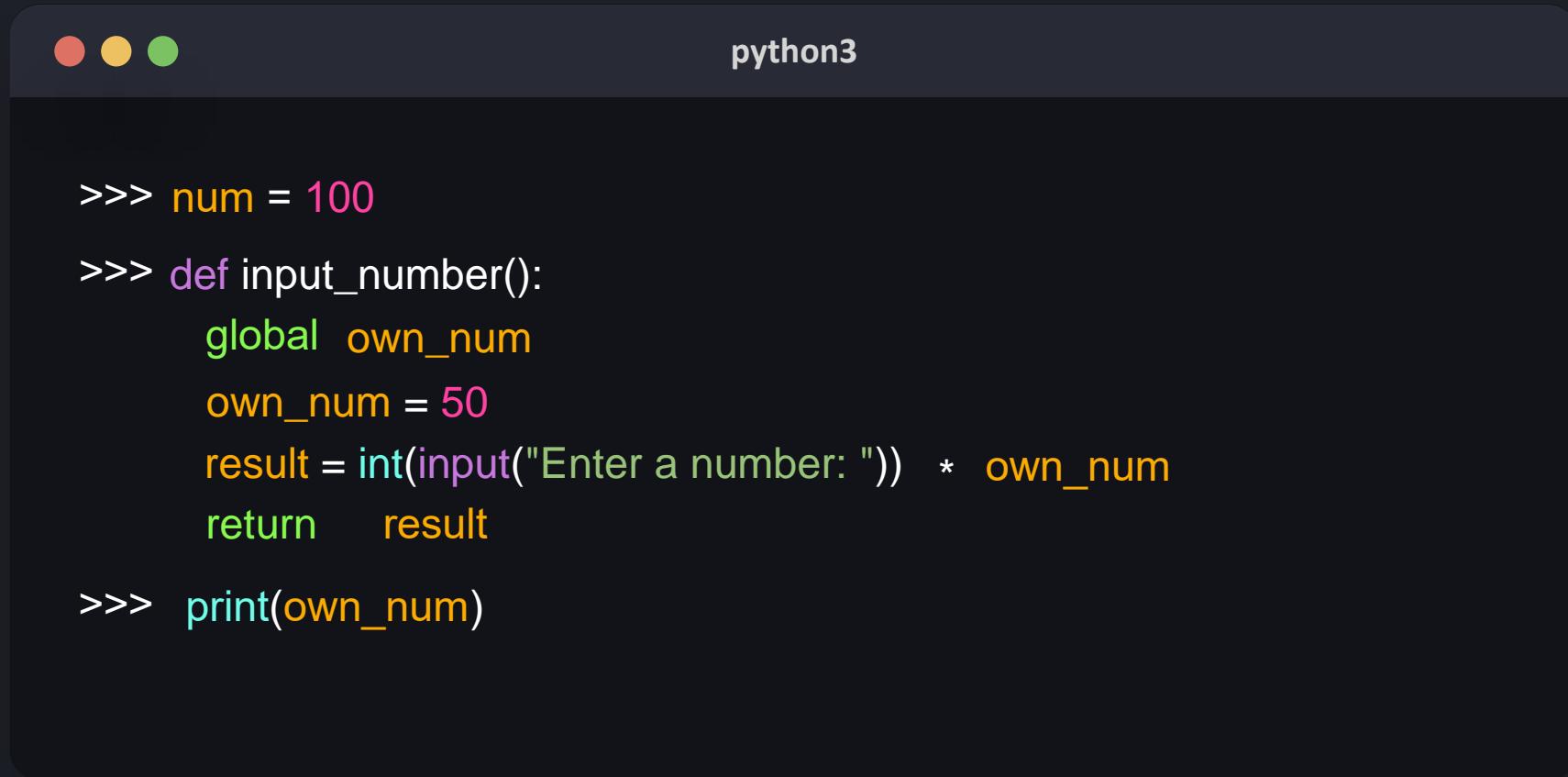
```
>>> num = 100
>>> def input_number():
    own_num = 50
    result = int(input("Enter a number: "))
    return result
>>> print(own_num)
```

The final line of code, `print(own_num)`, results in the following error message:

NameError: name 'own_num' is not defined

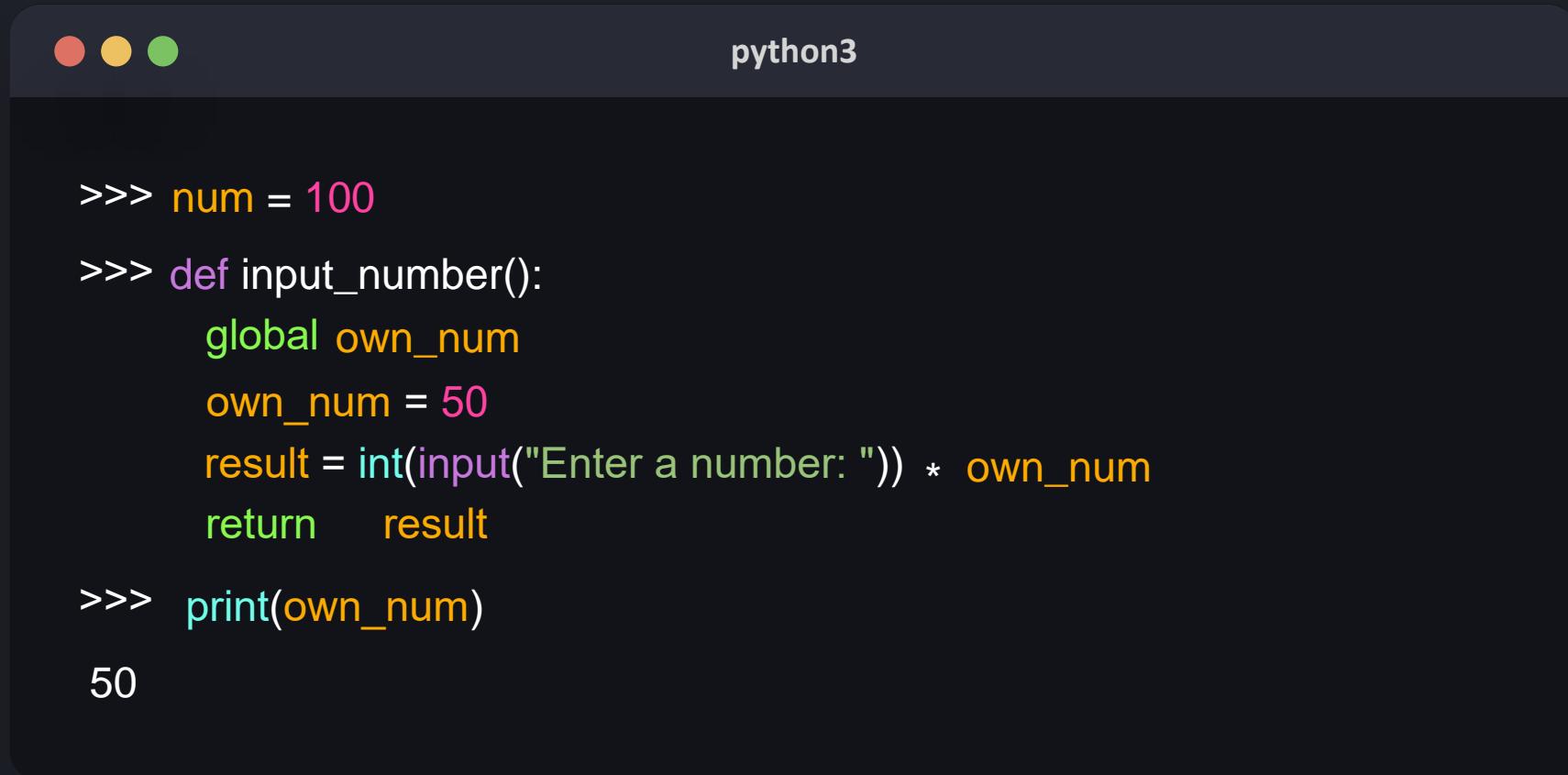


```
python3
>>> num = 100
>>> def input_number():
    global
    own_num = 50
    result = int(input("Enter a number: "))
    return result
           * own_num
           m
```



The image shows a screenshot of a macOS terminal window. The window title is "python3". Inside the terminal, the following Python code is displayed:

```
>>> num = 100
>>> def input_number():
    global own_num
    own_num = 50
    result = int(input("Enter a number: ")) * own_num
    return result
>>> print(own_num)
```



A screenshot of a macOS terminal window titled "python3". The window has a dark theme with red, yellow, and green close buttons. The code inside the terminal is as follows:

```
>>> num = 100
>>> def input_number():
    global own_num
    own_num = 50
    result = int(input("Enter a number: ")) * own_num
    return result
>>> print(own_num)
50
```

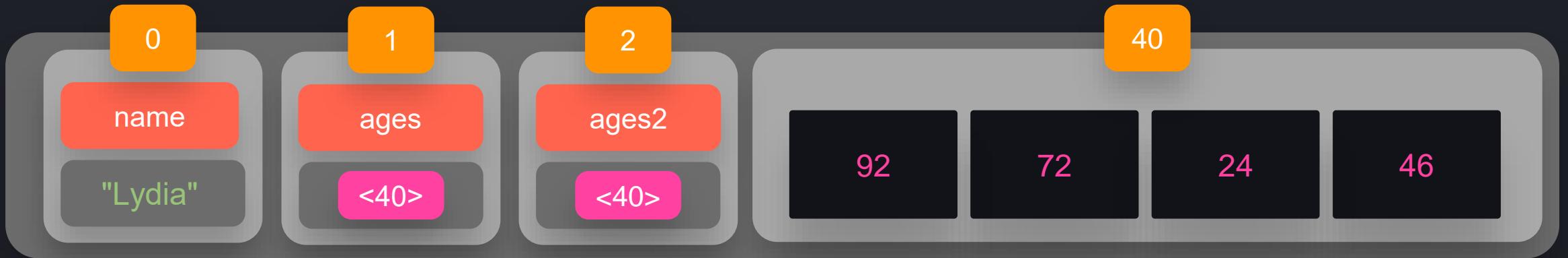


KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Function - Arguments Explained



python3

```
>>> name = "Lydia"
>>> ages = [56, 72, 24, 46]
>>> ages2 = ages
>>> ages[0] = 92
>>> print(ages2[0])
92
```



python3

```
>>> age = 22
>>> def multiply(          nu
                    hu        ):
                    *=  2      m
                    mprint("In multiply: ", str(num))
```



python3

```
>>> age = 22
>>> def multiply(      nu
                     hu   ):      )
                  *=  2      m
                  mprint("In multiply: ", str(num))
>>> multiply(age)
```



python3

```
>>> age = 22
>>> def multiply(          nu
                     hu        ):
                     *= 2      m
                     mprint("In multiply: ", str(num))

>>> multiply(age)
In multiply: 44
```



python3

```
>>> age = 22
>>> def multiply(          nu
                     hu        ):           )
                  *=   2      m
                     mprint("In multiply: ", str(num))

>>> multiply(age)
In multiply: 44

>>> print(age)
```



python3

```
>>> age = 22
>>> def multiply(          nu
                      hu        ):           m
                           *= 2      m
                           print("In multiply: ", str(num))

>>> multiply(age)
In multiply: 44

>>> print(age)
22
```



python3

```
>>> age = 22
>>> def multiply(    nu):
        hu      ): 
            *= 2      m
            mprint("In multiply: ", str(num))
>>> multiply(age)
```



python3

```
>>> age = 22
>>> def multiply(      nu
                     hu   ):      m
                         *= 2      m
                         print("In multiply: ", str(num))
>>> multiply(age)
```



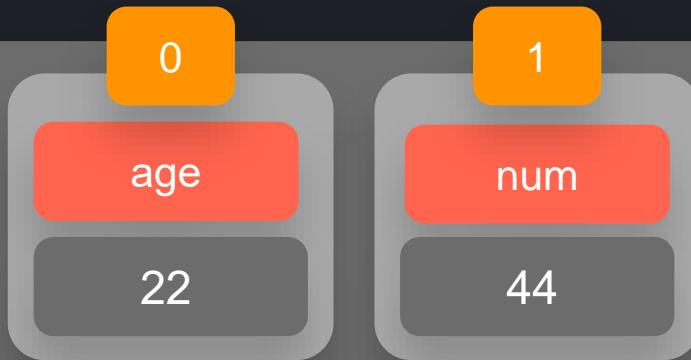
python3

```
>>> age = 22
>>> def multiply(      nu
                     hu   ):      m
                         *= 2      m
                         print("In multiply: ", str(num))
>>> multiply(age)
```



A screenshot of a terminal window titled "python3". The window has three colored status indicators (red, yellow, green) in the top-left corner. The terminal displays the following Python session:

```
>>> age = 22
>>> def multiply(num):
...     num *= 2
...     print("In multiply: ", str(num))
...
>>> multiply(age)
In multiply: 44
```



python3

```
>>> age = 22
>>> def multiply(      nu
                     hu   ):      m
                         *= 2      m
                     print("In multiply: ", str(num))

>>> multiply(age)
In multiply: 44

>>> print(age)
22
```



python3

```
>>> nums = [1, 2, 3]
>>> def change_first_item(list):
    list[0] = 9
```



python3

```
>>> nums = [1, 2, 3]
>>> def change_first_item(list):
    list[0] = 9
```



python3

```
>>> nums = [1, 2, 3]
>>> def change_first_item(list):
    list[0] = 9
```



python3

```
>>> nums = [1, 2, 3]
>>> def change_first_item(list):
    list[0] = 9
>>> change_first_item(nums)
```



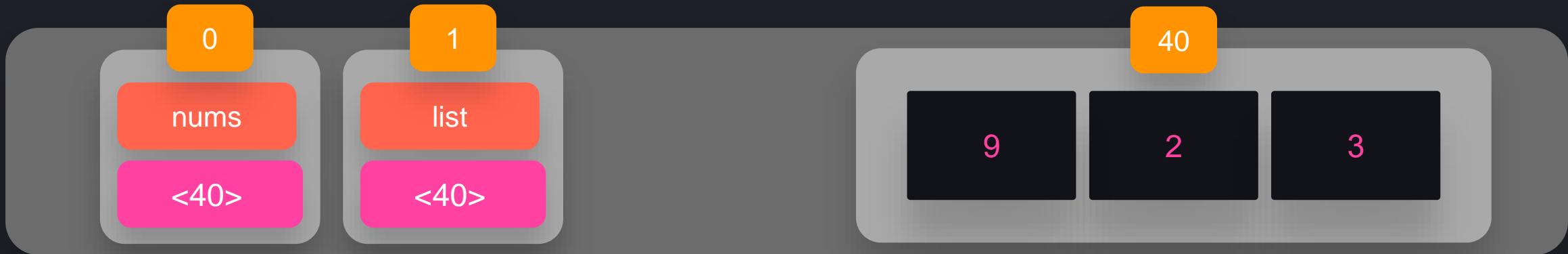
python3

```
>>> nums = [1, 2, 3]
>>> def change_first_item(list):
    list[0] = 9
>>> change_first_item(nums)
```



python3

```
>>> nums = [1, 2, 3]
>>> def change_first_item(list):
    list[0] = 9
>>> change_first_item(nums)
```



A screenshot of a terminal window titled "python3". The window has three status indicators at the top left: a red dot, a yellow dot, and a green dot. The terminal displays the following Python code:

```
>>> nums = [1, 2, 3]
>>> def change_first_item(list):
    list[0] = 9
>>> change_first_item(nums)
>>> print(nums)
```

The output of the code is [9, 2, 3], demonstrating that the mutation of the list in the function changes the original list variable.



python3

```
>>> nums = [1, 2, 3]
>>> def change_first_item(list):
    list[0] = 9
>>> change_first_item(nums)
>>> print(nums)
[9, 2, 3]
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Tuples

[1, 2, 3]

```
[1, 2, 3]
```

```
del myList[1]
```

```
myList.append(4)
```



python3

```
>>> tuple1 = (1, 2, 3)
```



python3

```
>>> tuple1 = (1, 2, 3)
>>> print(tuple1)
```



python3

```
>>> tuple1 = (1, 2, 3)
>>> print(tuple1)
(1, 2, 3)
```



python3

```
>>> tuple1 = (1, 2, 3)
>>> print(tuple1)
(1, 2, 3)

>>> tuple2 = 1, 2, 3
```



python3

```
>>> tuple1 = (1, 2, 3)
>>> print(tuple1)
(1, 2, 3)

>>> tuple2 = 1, 2, 3
>>> print(tuple2)
```



python3

```
>>> tuple1 = (1, 2, 3)
```

```
>>> print(tuple1)
```

```
(1, 2, 3)
```

```
>>> tuple2 = 1, 2, 3
```

```
>>> print(tuple2)
```

```
(1, 2, 3)
```



python3

```
>>> tuple1 = (1, 2, 3)
```



python3

```
>>> tuple1 = (1, 2, 3)
>>> for item in tuple1:
    print(item)
```



python3

```
>>> tuple1 = (1, 2, 3)
>>> for item in tuple1:
    print(item)

1
2
3
```



python3

```
>>> tuple1 = (1, 2, 3)
>>> for item in tuple1:
    print(item)

1
2
3

>>> print(tuple1[0:1])
```



python3

```
>>> tuple1 = (1, 2, 3)
>>> for item in tuple1:
    print(item)

1
2
3

>>> print(tuple1[0:1])
(0, 1)
```



python3

```
>>> tuple1 = (1, 2, 3)
```



python3

```
>>> tuple1 = (1, 2, 3)
>>> tuple1.append(4)
```



python3

```
>>> tuple1 = (1, 2, 3)
>>> tuple1.append(4)
```

AttributeError: 'tuple' object has no attribute 'append'



python3

```
>>> tuple1 = (1, 2, 3)
```

```
>>> tuple1.append(4)
```

```
AttributeError: 'tuple' object has no attribute 'append'
```

```
>>> tuple1[4] = 9
```

```
TypeError: 'tuple' object does not support item assignment
```



python3

```
>>> tuple1 = (1, 2, 3)
>>> tuple1.append(4)
```

AttributeError: 'tuple' object has no attribute 'append'

```
>>> tuple1[4] = 9
```

TypeError: 'tuple' object does not support item assignment

```
>>> del tuple1[1]
```

TypeError: 'tuple' object doesn't support item deletion



python3

```
>>> tuple1 = (1, 2, 3)
```

```
>>> tuple1.append(4)
```

```
AttributeError: 'tuple' object has no attribute 'append'
```

```
>>> tuple1[4] = 9
```

```
TypeError: 'tuple' object does not support item assignment
```

```
>>> del tuple1[1]
```

```
TypeError: 'tuple' object doesn't support item deletion
```



python3

```
>>> age = 22
>>> tuple1 = (1, "Lydia", age, (1, 2))
```



python3

```
>>> tuple1 = (1,)  
>>> tuple2 = 1,
```



KodeKloud

Check out our full course on Python Entry-Level Programmer Certification (PCEP): <https://kode.wiki/3NnDTIU>



Dictionaries

Name	Username
lydia	lydiahallie
sarah	sarah123
max	max
joe	joejoe



python3

>>>

```
"lydia": "lydiahallie",
"Sarah": "sarah123",
"max": "max_",
"joe": "joejoe",
}
```

Name

Username

lydia

lydiahallie

sarah

sarah123

max

max_

joe

joejoe



python3

>>>

```
usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}
```



python3

>>>

```
usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}
```

```
>>> print(usernames["sarah"])  
"sarah123"
```



python3

```
>>>  
  
usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
>>>} print(usernames["anotherone"])
```



python3

```
>>>  
usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
>>> print(usernames["anotherone"])  
KeyError: 'anotherone'
```

Methods

dictionary.keys()

dictionary.values()

dictionary.items()



python3

```
>>>  
usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
>>> print(usernames.keys())
```



python3

```
>>>
usernames = {
    "lydia": "lydiahallie",
    "sarah": "sarah123",
    "max": "max_",
    "joe": "joejoe",
}

>>> print(usernames.keys())
dict_keys(['lydia', 'sarah', 'max', 'joe'])
```



python3

```
>>>
usernames = {
    "lydia": "lydiahallie",
    "sarah": "sarah123",
    "max": "max_",
    "joe": "joejoe",
}
>>> print(usernames.keys())
dict_keys(['lydia', 'sarah', 'max', 'joe'])
```



python3

```
>>> usernames = {
    "lydia": "lydiahallie",
    "sarah": "sarah123",
    "max": "max_",
    "joe": "joejoe",
}

>>> for key in usernames.keys():
    print(key + " - " + usernames[key])
```



python3

```
>>> usernames = {
    "lydia": "lydiahallie",
    "sarah": "sarah123",
    "max": "max_",
    "joe": "joejoe",
}

>>> for key in usernames.keys():
    print(key + " - " + usernames[key])

lydia - lydiahallie
sarah - sarah123
max - max_
joe - joejoe
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> print(usernames.values())
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> print(usernames.values())  
  
dict_values(['lydiahallie', 'sarah123', 'max_', 'joejoe'])
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> print(usernames.items())
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> print(usernames.items())  
dict_items([  
    ('lydia', 'lydiahallie'),  
    ('sarah', 'sarah123'),  
    ('max', 'max_')  
    ('joe', 'joejoe')  
])
```

Methods

dictionary.keys()

dictionary.values()

dictionary.items()



python3

>>>

```
usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> usernames["max"] = "max123"
```



python3

```
>>>

usernames = {
    "lydia": "lydiahallie",
    "sarah": "sarah123",
    "max": "max_",
    "joe": "joejoe",

}
>>> usernames["max"] = "max123"
>>> print(usernames["max"])
```



python3

>>>

```
usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}
```

```
>>> usernames["max"] = "max123"
```

```
>>> print(usernames["max"])  
"max123"
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> usernames.update({ "chloe": "chloe123" })
```



python3

```
>>>
    usernames = {
        "lydia": "lydiahallie",
        "sarah": "sarah123",
        "max": "max_",
        "joe": "joejoe",
    }
>>> usernames.update({ "chloe": "chloe123" })
>>> print(usernames)
```



python3

```
>>>
usernames = {
    "lydia": "lydiahallie",
    "sarah": "sarah123",
    "max": "max_",
    "joe": "joejoe",
}
>>> usernames.update({ "chloe": "chloe123" })
>>> print(usernames)
{
    "lydia": "lydiahallie",
    "sarah": "sarah123",
    "max": "max_",
    "joe": "joejoe",
    "chloe": "chloe123"
}
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> del usernames["max"]
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> del usernames["max"]  
  
>>> print(usernames)
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> del usernames["max"]  
  
>>> print(usernames)  
{  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "joe": "joejoe"  
}
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> usernames.clear()
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> usernames.clear()  
>>> print(usernames)
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> usernames.clear()  
>>> print(usernames)  
{}
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> usernames.popitem()  
>>> print(usernames)
```



python3

```
>>> usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
  
>>> usernames.popitem()  
  
>>> print(usernames)  
{  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_"  
}
```



python3

```
>>>  
  
usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
>>> usernames_copy = usernames.copy()
```



python3

```
>>>  
usernames = {  
    "lydia": "lydiahallie",  
    "sarah": "sarah123",  
    "max": "max_",  
    "joe": "joejoe",  
}  
>>> usernames_copy = usernames.copy()  
>>> print(usernames_copy)
```



python3

```
>>>
    usernames = {
        "lydia": "lydiahallie",
        "sarah": "sarah123",
        "max": "max_",
        "joe": "joejoe",
    }
>>> usernames_copy = usernames.copy()
>>> print(usernames_copy)
{
    "lydia": "lydiahallie",
    "sarah": "sarah123",
    "max": "max_",
    "joe": "joejoe"
}
```



KodeKloud

Learn more about DevOps and Cloud courses with KodeKloud: <https://kode.wiki/3N3A4kt>