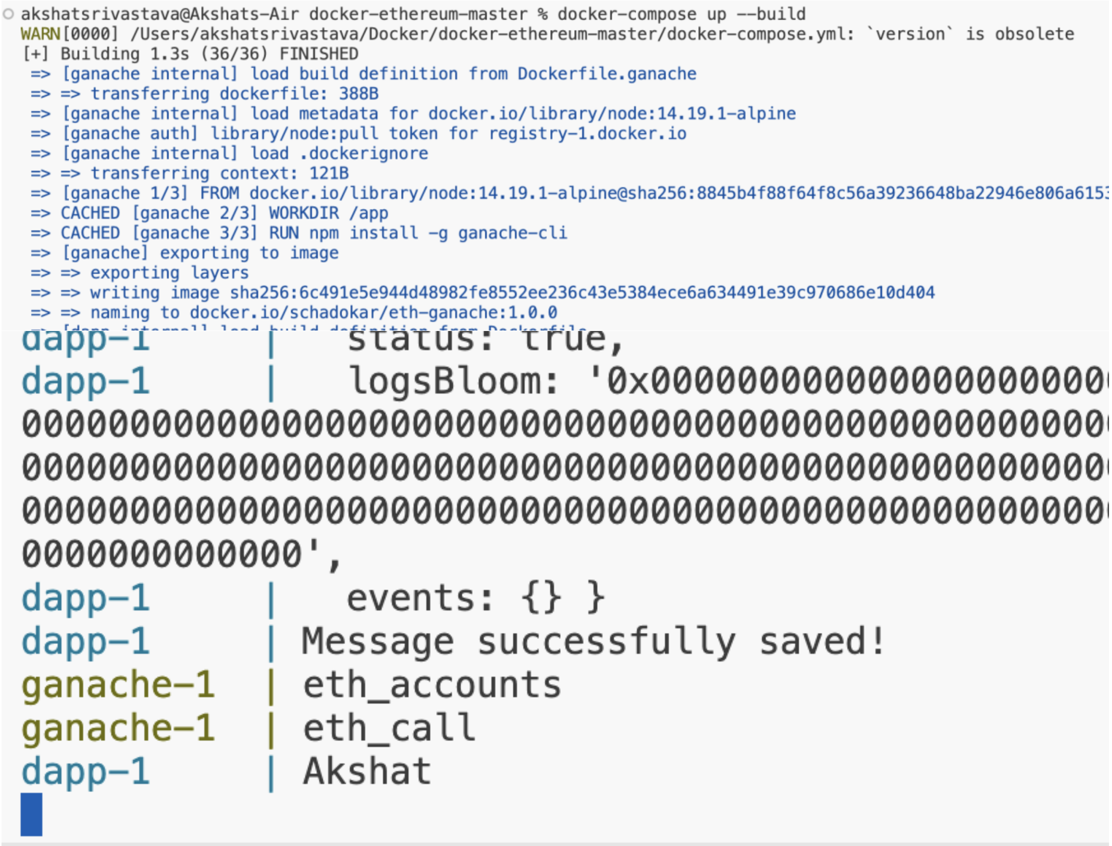
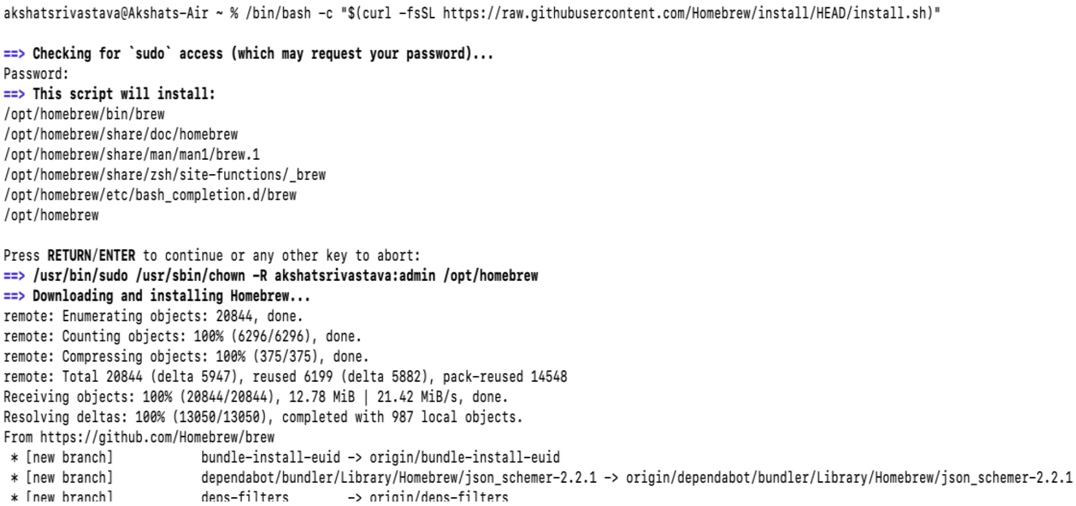
Assignment 2

1. Run full-stack Ethereum using Docker containers.

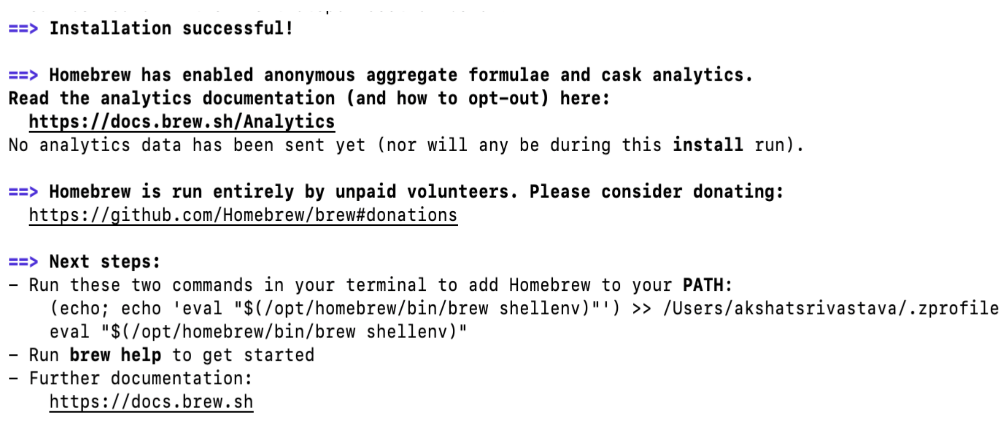


```
○ akshatsrivastava@Akshats-Air docker-ethereum-master % docker-compose up --build
  WARN[0000] /Users/akshatsrivastava/Docker/docker-ethereum-master/docker-compose.yml: `version` is obsolete
  [+] Building 1.3s (36/36) FINISHED
   => [ganache internal] load build definition from Dockerfile.ganache
   => => transferring dockerfile: 388B
   => [ganache internal] load metadata for docker.io/library/node:14.19.1-alpine
   => [ganache auth] library/node:pull token for registry-1.docker.io
   => [ganache internal] load .dockerignore
   => => transferring context: 121B
   => [ganache 1/3] FROM docker.io/library/node:14.19.1-alpine@sha256:8845b4f88f64f8c56a39236648ba22946e806a615
   => CACHED [ganache 2/3] WORKDIR /app
   => CACHED [ganache 3/3] RUN npm install -g ganache-cli
   => [ganache] exporting to image
   => => exporting layers
   => => writing image sha256:6c491e5e944d48982fe8552ee236c43e5384ece6a634491e39c970686e10d404
   => => naming to docker.io/schadokar/eth-ganache:1.0.0
   => [dapp internal] load build definition from Dockerfile
  dapp-1       |    status: true,
  dapp-1       |    logsBloom: '0x0000000000000000000000
  0000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000
  0000000000000000000000000000000000000000000000000
  0000000000000',
  dapp-1       |    events: {} }
  dapp-1       |  Message successfully saved!
  ganache-1    |  eth_accounts
  ganache-1    |  eth_call
  dapp-1       |  Akshat
```

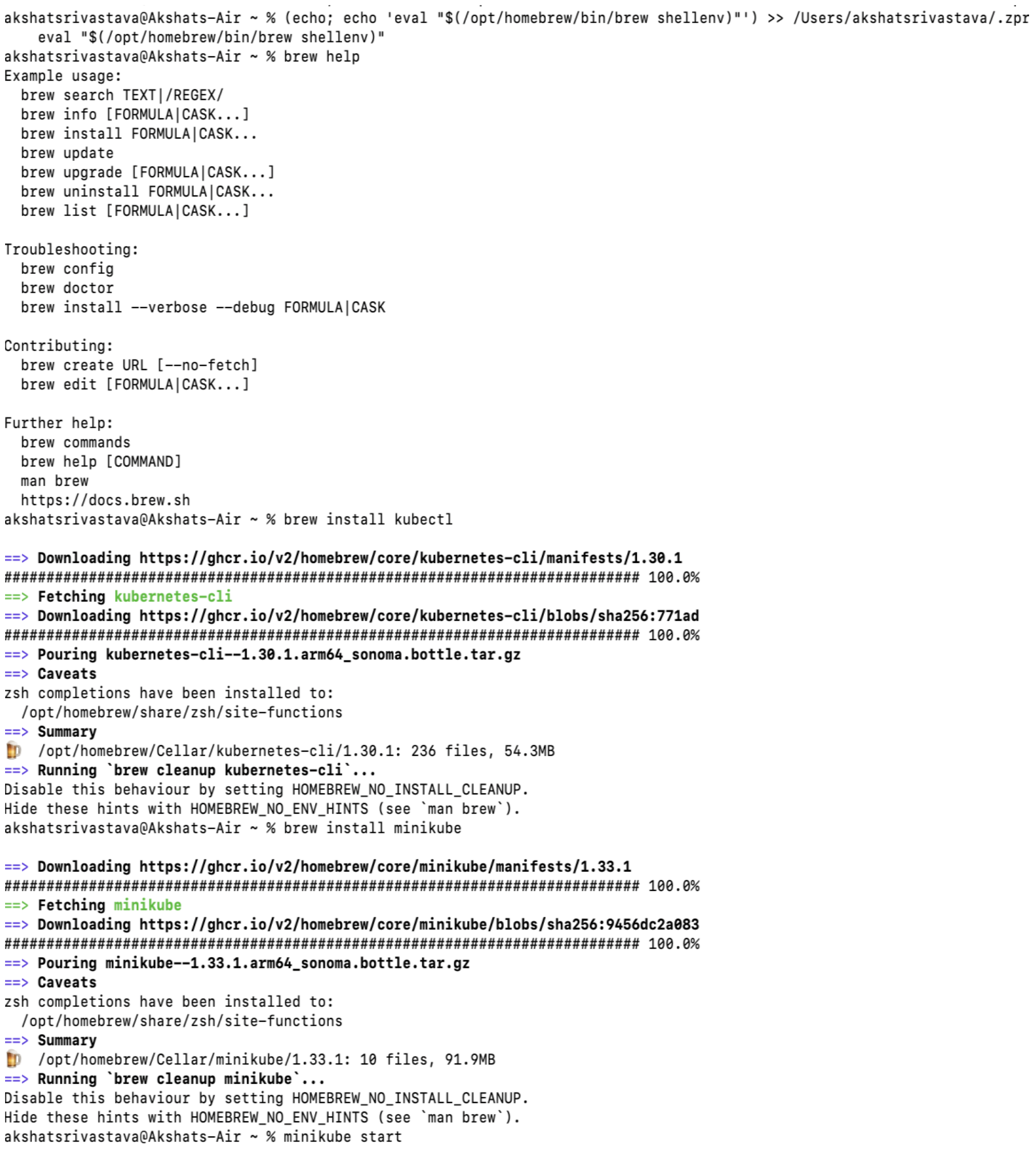
2. Install kubernetes (kubectl and minikube)



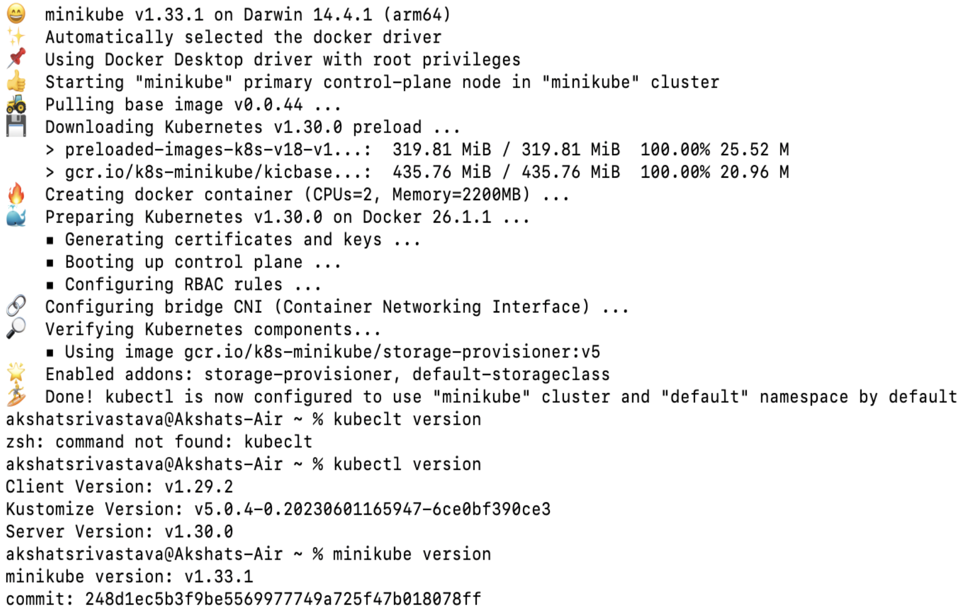
```
akshatsrivastava@Akshats-Air ~ % /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

==> Checking for `sudo` access (which may request your password)...
Password:
==> This script will install:
/opt/homebrew/bin/brew
/opt/homebrew/share/doc/homebrew
/opt/homebrew/share/man/man1/brew.1
/opt/homebrew/share/zsh/site-functions/_brew
/opt/homebrew/etc/bash_completion.d/brew
/opt/homebrew

Press RETURN/ENTER to continue or any other key to abort:
==> /usr/bin/sudo /usr/sbin/chown -R akshatsrivastava:admin /opt/homebrew
==> Downloading and installing Homebrew...
remote: Enumerating objects: 20844, done.
remote: Counting objects: 100% (6296/6296), done.
remote: Compressing objects: 100% (375/375), done.
remote: Total 20844 (delta 5947), reused 6199 (delta 5882), pack-reused 14548
Receiving objects: 100% (20844/20844), 12.78 MiB | 21.42 MiB/s, done.
Resolving deltas: 100% (13050/13050), completed with 987 local objects.
From https://github.com/Homebrew/brew
 * [new branch]          bundle-install-euid -> origin/bundle-install-euid
 * [new branch]          dependabot/bundler/Library/Homebrew/json_schemer-2.2.1 -> origin/dependabot/bundler/Library/Homebrew/json_schemer-2.2.1
 * [new branch]          deps-filters        -> origin/deps-filters
```



```
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
    (echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"') >> /Users/akshatsrivastava/.zprofile
    eval "$(/opt/homebrew/bin/brew shellenv)"
- Run brew help to get started
- Further documentation:
    https://docs.brew.sh
```



```
akshatsrivastava@Akshats-Air ~ % (echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"') >> /Users/akshatsrivastava/.zpr
    eval "$(/opt/homebrew/bin/brew shellenv)"
akshatsrivastava@Akshats-Air ~ % brew help
Example usage:
  brew search TEXT|/REGEX/
  brew info [FORMULA|CASK...]
  brew install FORMULA|CASK...
  brew update
  brew upgrade [FORMULA|CASK...]
  brew uninstall FORMULA|CASK...
  brew list [FORMULA|CASK...]

Troubleshooting:
  brew config
  brew doctor
  brew install --verbose --debug FORMULA|CASK

Contributing:
  brew create URL [--no-fetch]
  brew edit [FORMULA|CASK...]

Further help:
  brew commands
  brew help [COMMAND]
  man brew
  https://docs.brew.sh
akshatsrivastava@Akshats-Air ~ % brew install kubectl

==> Downloading https://ghcr.io/v2/homebrew/core/kubernetes-cli/manifests/1.30.1
############################################################################ 100.0%
==> Fetching kubernetes-cli
==> Downloading https://ghcr.io/v2/homebrew/core/kubernetes-cli/blobs/sha256:771ad
############################################################################ 100.0%
==> Pouring kubernetes-cli--1.30.1.arm64_sonoma.bottle.tar.gz
==> Caveats
zsh completions have been installed to:
  /opt/homebrew/share/zsh/site-functions
==> Summary
🍺  /opt/homebrew/Cellar/kubernetes-cli/1.30.1: 236 files, 54.3MB
==> Running `brew cleanup kubernetes-cli`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
akshatsrivastava@Akshats-Air ~ % brew install minikube

==> Downloading https://ghcr.io/v2/homebrew/core/minikube/manifests/1.33.1
############################################################################ 100.0%
==> Fetching minikube
==> Downloading https://ghcr.io/v2/homebrew/core/minikube/blobs/sha256:9456dc2a083
############################################################################ 100.0%
==> Pouring minikube--1.33.1.arm64_sonoma.bottle.tar.gz
==> Caveats
zsh completions have been installed to:
  /opt/homebrew/share/zsh/site-functions
==> Summary
🍺  /opt/homebrew/Cellar/minikube/1.33.1: 10 files, 91.9MB
==> Running `brew cleanup minikube`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
akshatsrivastava@Akshats-Air ~ % minikube start
```



```
😄  minikube v1.33.1 on Darwin 14.4.1 (arm64)
✨  Automatically selected the docker driver
📌  Using Docker Desktop driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.44 ...
💾  Downloading Kubernetes v1.30.0 preload ...
    > preloaded-images-k8s-v18-v1...:  319.81 MiB / 319.81 MiB  100.00% 25.52 M
    > gcr.io/k8s-minikube/kicbase...:  435.76 MiB / 435.76 MiB  100.00% 20.96 M
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
akshatsrivastava@Akshats-Air ~ % kubeclt version
zsh: command not found: kubeclt
akshatsrivastava@Akshats-Air ~ % kubectl version
Client Version: v1.29.2
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
Server Version: v1.30.0
akshatsrivastava@Akshats-Air ~ % minikube version
minikube version: v1.33.1
commit: 248d1ec5b3f9be5569977749a725f47b018078ff
```

3. Quiz

Kubernetes is a robust platform for automating the deployment, scaling, and management of containerized applications. Its core functionality includes intelligent workload scheduling, allowing containers to be distributed efficiently across clusters to optimize resource utilization and ensure high availability. Kubernetes also offers features for horizontal scaling, service discovery, and load balancing, empowering organizations to handle varying levels of traffic seamlessly while maintaining reliability and performance.

In addition to basic orchestration, Kubernetes facilitates advanced deployment strategies like rolling updates and rollbacks, streamlining the process of releasing new application versions without disrupting service availability. Its self-healing capabilities automatically detect and recover from container failures, enhancing application resilience. Furthermore, Kubernetes supports multi-cloud and hybrid cloud deployments, providing a consistent platform for managing applications across diverse infrastructure environments. Overall, Kubernetes simplifies the complexities of container management, enabling teams to focus on innovation and delivering value to users.