

2.

```

macbook@ulugbek-2 ~ % brew install kubectll
Running 'brew update --auto-update'...
==> Auto-updated Homebrew!
Updated 5 taps (homebrew/bundle, homebrew/services, mongodb/brew, homebrew/core and homebrew/cask).
==> New Formulae
c3c cargo-llvm-cov csviewns
doltgres dotter help
helm-ls hopscootch-map jot
k8sgpt kiota libnsbpm
libwpcaplet ncndump netsurf-buildsystem
pivrit rathole rattler-build
ruby@3.2 steampguard-clif sugarjar
texi2mdoc urlscan vellid
vulkan-volk wasmedge zigmod
zipkin
==> New Casks
aqua blockstream-green ccreality-print egovframedev
emby geekbench-m1 heynote ia-presenter
imazing-profile-editor jupytering keyboard-cowboy lightburn
lw-scanner markedit ollama opencat
openstheasaurus-deutsch pile prettylean roan
sakura salt shadow-bot streammusic
taccy waketime xact znote
You have 5 outdated formulae installed.
==> Downloading https://ghcr.io/v2/homebrew/core/kubernetes-cli/manifests/1.29.0
==> Fetching kubernetes-cli
==> Downloading https://ghcr.io/v2/homebrew/core/kubernetes-cli/blobs/sha256:342175c8b07a09270054b036e0b06bcha56871d8291f96c9e4564bdabad72104
==> Pouring kubernetes-cli--1.29.0.arm64_sonoma.bottle.tar.gz
==> Caveats
zsh completions have been installed to:
  /opt/homebrew/share/zsh/site-functions
==> Summary
  /opt/homebrew/Cellar/kubernetes-cli/1.29.0: 234 files, 59.2MB
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').
macbook@ulugbek-2 ~ % curl -LO "https://dl.k8s.io/release/stable.txt" -s https://dl.k8s.io/release/stable.txt | bin/darwin/amd64/kubectl
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 138 100 138 0 0 2788 0 --:--:-- --:--:-- --:--:-- 3136
100 53.2M 100 53.2M 0 0 58.3M 0 --:--:-- --:--:-- --:--:-- 30.3M
macbook@ulugbek-2 ~ % sudo mv ./kubectl /usr/local/bin/kubectl
sudo chown root: /usr/local/bin/kubectl
Password:
macbook@ulugbek-2 ~ % kubectl version --client
Client Version: v1.29.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
macbook@ulugbek-2 ~ %
macbook@ulugbek-2 ~ % curl -LO "https://dl.k8s.io/release/stable.txt" -s https://dl.k8s.io/release/stable.txt | bin/darwin/amd64/kubectl
sudo install -o root -g wheel -m 0755 kubectl /usr/local/bin/kubectl
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 138 100 138 0 0 2874 0 --:--:-- --:--:-- --:--:-- 3136
100 53.2M 100 53.2M 0 0 63.5M 0 --:--:-- --:--:-- --:--:-- 71.0M
Password:
macbook@ulugbek-2 ~ % curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-darwin-amd64
sudo install minikube-darwin-amd64 /usr/local/bin/minikube
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 88.1M 100 88.1M 0 0 39.8M 0 0:00:02 0:00:02 --:--:-- 39.8M
macbook@ulugbek-2 ~ % minikube start

You are trying to run the amd64 binary on an M1 system.
Please consider running the darwin/arm64 binary instead.
Download at https://github.com/kubernetes/minikube/releases/download/v1.32.0/minikube-darwin-arm64

minikube v1.32.0 on Darwin 14.1.2
* Automatically selected the docker driver
* Using Docker Desktop driver with root privileges
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.28.3 preload ...
  > preloaded-images-k8s-v18-v1...: 341.16 MiB / 341.16 MiB 100.00% 35.21 M
  > gcr.io/k8s-minikube/kicbase...: 453.90 MiB / 453.90 MiB 100.00% 42.63 M
* Creating docker container (CPUs=2, Memory=4000MB) ...
* Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
  * 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
macbook@ulugbek-2 ~ %

```

3.

1: Kubernetes primarily orchestrates containerized applications, focusing on deploying, managing, and scaling them. It doesn't deploy or build your application's source code, nor does it offer built-in application services like databases or messaging systems.

2: Other notable orchestration tools besides Kubernetes include Docker Swarm, Apache Mesos, OpenShift, Nomad, and Rancher. Each offers unique features and is suited to different operational requirements or environments.