Development environment

This page will go over setting up your development environment to run Celestia software. This environment can be used for development, building binaries, and running nodes.

Install dependencies

1. If you are on Ubuntu, first update and upgrade your OS:
2. APT
3. YUM
4. bash
5. sudo
6. apt
7. update
8. &&
9. sudo
10. apt
11. upgrade
12. -y
13. sudo
14. apt
15. update
16. &&
17. sudo
18. apt
19. upgrade
20. -y
21. bash
22. sudo
23. yum
24. update
25. sudo
26. yum
27. update
28. Install essential packages that are necessary to execute many tasks like downloading files, compiling, and monitoring the node:
29. APT
30. YUM
31. Mac
32. bash
33. sudo
34. apt
35. install
36. curl
37. tar
38. wget
39. aria2
40. clang
41. pkg-config
42. libssl-dev
43. jq
44. build-essential
45. \
46. git
47. make
48. ncdu
49. -y
50. sudo
51. apt
52. install
53. curl
54. tar
55. wget
56. aria2
57. clang
58. pkg-config
59. libssl-dev

jq
build-essential
\
git
make
ncdu
-y
bash
sudo
yum
install
curl
tar
wget
aria2
clang
pkg-config
libssl-dev
jq
build-essential
\
git
make
ncdu
-y
sudo
yum
install
curl
tar
wget
aria2
clang
pkg-config
libssl-dev
jq
build-essential
\
git
make
ncdu
-y
bash

these commands are for installing Homebrew, wget and jq

follow the instructions from the output after running this command

/bin/bash
-c
"(
curl
-fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

then install wget & jq

brew
install
wget
&&

116. brew
117. install
118. jq

119. these commands are for installing Homebrew, wget and jq

120. follow the instructions from the output after running this command

121. /bin/bash
122. -c
123. "(
124. curl
125. -fsSL
126. https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

127. then install wget & jq

128. brew
129. install
130. wget
131. &&
132. brew
133. install
134. jq

Install Golang

celestia-node is written in Golang so we must install Golang to build and run our node.

1. Set the version for your desired network:
2. Mainnet Beta
3. Mocha
4. Arabica
5. bash
6. ver
7. =
8. "1.23.0"
9. ver
10. =
11. "1.23.0"
12. bash
13. ver
14. =
15. "1.23.2"
16. ver
17. =
18. "1.23.2"
19. bash
20. ver
21. =
22. "1.23.2"
23. ver
24. =
25. "1.23.2"
26. Download and install Golang:
27. Ubuntu (AMD)
28. Ubuntu (ARM)
29. Mac (Apple)
30. Mac (Intel)
31. bash
32. cd

33. HOME
34. wget
35. "https://golang.org/dl/go
36. ver
37. .linux-amd64.tar.gz"
38. sudo
39. rm
40. -rf
41. /usr/local/go
42. sudo
43. tar
44. -C
45. /usr/local
46. -xzf
47. "go
48. ver
49. .linux-amd64.tar.gz"
50. rm
51. "go
52. ver
53. .linux-amd64.tar.gz"
54. cd
55. HOME
56. wget
57. "https://golang.org/dl/go
58. ver
59. .linux-amd64.tar.gz"
60. sudo
61. rm
62. -rf
63. /usr/local/go
64. sudo
65. tar
66. -C
67. /usr/local
68. -xzf
69. "go
70. ver
71. .linux-amd64.tar.gz"
72. rm
73. "go
74. ver
75. .linux-amd64.tar.gz"
76. bash
77. cd
78. HOME
79. wget
80. "https://golang.org/dl/go
81. ver
82. .linux-arm64.tar.gz"
83. sudo
84. rm
85. -rf
86. /usr/local/go
87. sudo
88. tar
89. -C
90. /usr/local
91. -xzf
92. "go
93. ver
94. .linux-arm64.tar.gz"
95. rm
96. "go
97. ver
98. .linux-arm64.tar.gz"
99. cd
100. HOME

wget
"https://golang.org/dl/go
ver
.linux-arm64.tar.gz"
sudo
rm
-rf
/usr/local/go
sudo
tar
-C
/usr/local
-xzf
"go
ver
.linux-arm64.tar.gz"
rm
"go
ver
.linux-arm64.tar.gz"
bash
cd
HOME
wget
"https://golang.org/dl/go
ver
.darwin-arm64.tar.gz"
sudo
rm
-rf
/usr/local/go
sudo
tar
-C
/usr/local
-xzf
"go
ver
.darwin-arm64.tar.gz"
rm
"go
ver
.darwin-arm64.tar.gz"
cd
HOME
wget
"https://golang.org/dl/go
ver
.darwin-arm64.tar.gz"
sudo
rm
-rf
/usr/local/go
sudo
tar
-C
/usr/local
-xzf
"go
ver
.darwin-arm64.tar.gz"
rm
"go
ver
.darwin-arm64.tar.gz"
bash
cd
HOME

wget
"https://golang.org/dl/go
ver
.darwin-amd64.tar.gz"
sudo
rm
-rf
/usr/local/go
sudo
tar
-C
/usr/local
-xzf
"go
ver
.darwin-amd64.tar.gz"
rm
"go
ver
.darwin-amd64.tar.gz"
cd
HOME
wget
"https://golang.org/dl/go
ver
.darwin-amd64.tar.gz"
sudo
rm
-rf
/usr/local/go
sudo
tar
-C
/usr/local
-xzf
"go
ver
.darwin-amd64.tar.gz"
rm
"go
ver
.darwin-amd64.tar.gz"
Add your/usr/local/go/bin
directory to yourPATH
if you have not already:
bash
zsh
bash
echo
"export PATH=
PATH
:/usr/local/go/bin:
HOME
/go/bin"

HOME
/.bash_profile
source
HOME
/.bash_profile
echo
"export PATH=
PATH
:/usr/local/go/bin:
HOME
/go/bin"

HOME
/.bash_profile
source
HOME
/.bash_profile
bash
echo
"export PATH=
PATH
:/usr/local/go/bin:
HOME
/go/bin"

HOME
/.zshrc
source
HOME
/.zshrc
echo
"export PATH=
PATH
:/usr/local/go/bin:
HOME
/go/bin"

HOME
/.zshrc
source
HOME
/.zshrc
TIP
Useecho SHELL
to figure out what type of shell you are using!
To verify that the correct version of Go was installed correctly run:
bash
go
version
go
version

The output will show the version installed. [[Edit this page on GitHub] Last updated: Previous page Deciding which node to run Next page Install celestia-node []