# The Developed and Prepared Piece of True Chain-----Environment Building (Centos)

作者：邹伟 审阅：

This article is a supplement for the Environment Building.The CentOS environment is roughly the same as the Ubuntu environment, with slightly different details. So I wrote this article after repeated consideration. The environment uses Alibaba Cloud ECS, cloud service construction, so that everyone can quickly learn to master the CentOS environment configuration.

## Build A TrueChain Cloud Runtime Environment

## System version: Cent 7.\*

# tools  
# git  
# go => version 1.7 or higher  
# compilation tools

## Step 1: Install Git

First，checking if the system has been installed with Git.

$ git --version

If the version number is not displayed correctly, you will need to install Git.

Then, because the version of Git is not very high, so using the package management tools to install it directly .

$ sudo yum install git

Finally, after the installation, checking if the installation is successful.

$ git --version

fig:If the installation is correct, it will display Git version number.

## Step2: Install Go

First,checking if the system has been installed with Go.

$ go version

When not installed, there are three ways to install it: install it by using the package management tool, install it by using the officially released installation package, and install it by using source code.It is most convenient to install by using the package management tool, but if the installed version is low, you need to use the installation package to install.Please refer to Go official website for installation steps by using source code.（https://golang.org/doc/install/source）

### Package Management Tool Installation

$ sudo yum install golang

### Installation Package Installation

After testing, it is found that the latest version 1.11 of Go will report runtime error when compiling True Chain. The version of 1.10 is normal, so it is recommended to download version 1.10,Linux-1.10.4, 64-bit version. The download address is as follows, use wget to download directly.The cloud server does not need to bring the VPN(Virtual Private Network) function, so it saves the link of VPN.

# to download cloud service directly  
$ wget https://dl.google.com/go/go1.10.4.linux-amd64.tar.gz

If it is a local CentOS link, you will need the VPN function.

The official website download address：[https://golang.org/dl/](https://link.zhihu.com/?target=https%3A//golang.org/dl/).There are various versions of the address link on the official website, so it is not listed. In addition, since the cloud server can directly download, the way of mirror download is omitted.If you have any needs, please visit the help posts of other people to find a download package.

**Command-mode installation of Go**

Here directly take the Go1.10.4.linux-amd64.tar.gz file as an example, other versions can directly replace the file name.

Note: Installation location/usr/local\*\*

sudo tar -C /usr/local -xzf go1.10.4.linux-amd64.tar.gz // Note that the correct file name is entered.

**Step 3:Set Environment Variables**

First ,creating a GOPATH folder to save our working directory.If the server creates multiple accounts, you need to pay attention to the permissions issue and use the root account directly.And so cd open the root directory.

# enter your current user directory  
$ cd ~  
# create go working directory  
$ mkdir go  
# open go working directory  
$ cd go  
# create bin, pkg and src folders in go working directory  
$ mkdir bin pkg src

**Configure the Go working directory (temporary environment variables)**

# execute the temporary environment variable scenarios directly  
$ export PATH=$PATH:/usr/local/go/bin:$GOPATH/bin  
# set working directory  
$ export GOPATH=~/go

Note: Variables that are directly set using export are temporary variables, which means exiting the current shell and the values defined for variables will not take effect.How can we make the variables we define become permanently effective?

**Configure the Go working directory（ configure environment variables permanently ）**

# open the profile file to modify  
$ vi /etc/profile  
# add this configuration to the configuration file  
export PATH=$PATH:/usr/local/go/bin:$GOPATH/bin  
# configure working directory  
export GOPATH=~/go  
# Save the profile file and execute the following commands after the modification.It does not need to be restarted.  
$ source /etc/profile

Then using go version to see if the installation was successful.

Testing whether the installation is successful：

Writing the following code into $HOME/go/src/hello/hello.go

package main  
import "fmt"  
func main() {  
 fmt.Printf("hello, world\n")  
}

Then compiling and running：

$ cd $HOME/go/src/hello  
$ go build  
$ ./hello // export：hello,world

## Step 4：Install Compilation Environment

$ sudo yum groupinstall "Development Tools"

## Compile and Run TrueChain

Method 1 (recommended): Using the following command to import the initial source code in the console.

$ go get github.com/truechain/truechain-engineering-code

Then you will start downloading the source code, the source code will be downloaded to the previously configured GOPATH path.If an error is reported：fatal: unable to access 'https://github.com/truechain/truechain-engineering-code/': Peer reports incompatible or unsupported protocol version.It can be solved by the following command.

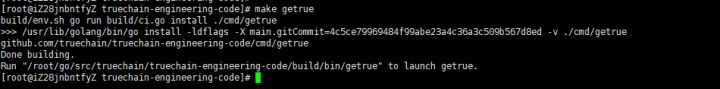
$ yum update -y nss curl libcurl

Method 2: Entering the $HOME/go/src/truechain directory.If there is no TrueChain directory, creating a mkdir truechain command to create it.Entering the TrueChain directory and execute the following command to clone the TrueChain project code.

$ git clone https://github.com/truechain/truechain-engineering-code.git

Entering the truechain-engineering-code directory after downloading through method one or method two：

make getrue // only compile getrue  
// or  
make all // compile all tools



Compiled successfully.

The executable file which is compiled successfully is in the build/bin/ directory,and running the corresponding file directly in this directory.

$ cd ./build/bin // enter the compilation directory

In order for the getrue command to run in any folder, the following operations are required.

$ cp getrue $GOPATH/bin // copy the compiled file to the bin directory of $GOPATH

$ getrue init ../../cmd/getrue/genesis.json // may report an error also available in any directory getrue init $GOPATH/src/github.com/truechain/truechain-engineering-code/cmd/getrue/genesis.json  
$ ./getrue --nodiscover --singlenode --mine --etherbase 0x8a45d70f096d3581866ed27a5017a4eeec0db2a1 --bftkeyhex c1581e25937d9ab91421a3e1a2667c85b0397c75a195e643109938e987acecfc --bftip 192.1  
68.68.43 --bftport 10080

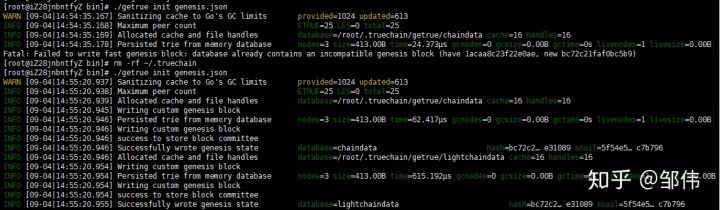
Initializing genesis.json may give an error.

Fatal: Failed to write fast genesis block: database already contains an incompatible genesis block (have 1acaa8c23f22e0ae, new bc72c21faf0bc5b9)

If this error occurs, deleting the TrueChain cache directory.

$ rm -rf ~/.truechain

Initialization and operation are successful. As shown below:



Initialized successfully.