1 在vimrc中添加  
Plug 'ycm-core/YouCompleteMe', { 'do': 'python3 ./install.py --clangd-completer --java-completer' }

2 通过PlugInstall命令在~/.vim/plugged文件夹下载源代码

# 3 遇到ycmd server SHUT DOWN问题

# 需要进行代码编译，但是编译时候没有下载完全 所以需要执行

git submodule update --init --recursive

# 4 保证代码下载完全后，进行编译遇到下载mishash的问题，手动下载到指定位置

# 5 如果不强制要求all支持，可以仅仅使用支持c和java语言的版本

6 Found /home/klpchan/.vim/plugged/YouCompleteMe/third\_party/ycmd/.ycm\_extra\_conf.py. Load?

" 不显示开启vim时检查ycm\_extra\_conf文件的信息

let g:ycm\_confirm\_extra\_conf = 0

" 开启基于tag的补全，可以在这之后添加需要的标签路径

let g:ycm\_collect\_identifiers\_from\_tags\_files = 1

" 开启语义补全

let g:ycm\_seed\_identifiers\_with\_syntax = 1

"注释和字符串中的文字也会被收入补全

let g:ycm\_collect\_identifiers\_from\_comments\_and\_strings = 0

" 输入第 2 个字符开始补全

let g:ycm\_min\_num\_of\_chars\_for\_completion= 2

" 禁止缓存匹配项,每次都重新生成匹配项

let g:ycm\_cache\_omnifunc=0

"在注释输入中也能补全

let g:ycm\_complete\_in\_comments = 1

"在字符串输入中也能补全

let g:ycm\_complete\_in\_strings = 1

"定义快捷健补全

let g:ycm\_key\_list\_select\_completion = ['<c-n>', '<Down>']

let g:ycm\_key\_list\_previous\_completion = ['<c-p>', '<Up>']

" 设置在下面几种格式的文件上屏蔽ycm

let g:ycm\_filetype\_blacklist = {

\ 'tagbar' : 1,

\ 'qf' : 1,

\ 'notes' : 1,

\ 'markdown' : 1,

\ 'unite' : 1,

\ 'text' : 1,

\ 'vimwiki' : 1,

\ 'pandoc' : 1,

\ 'infolog' : 1,

\ 'mail' : 1

\}

"设置关健字触发补全

let g:ycm\_semantic\_triggers = {

\ 'c' : ['->', '.', ' ', '(', '[', '&'],

\ 'objc' : ['->', '.', 're!\[[\_a-zA-Z]+\w\*\s', 're!^\s\*[^\W\d]\w\*\s',

\ 're!\[.\*\]\s'],

\ 'ocaml' : ['.', '#'],

\ 'cpp,objcpp' : ['->', '.', '::'],

\ 'perl' : ['->'],

\ 'php' : ['->', '::'],

\ 'cs,java,javascript,typescript,d,python,perl6,scala,vb,elixir,go' : ['.'],

\ 'ruby' : ['.', '::'],

\ 'lua' : ['.', ':'],

\ 'erlang' : [':'],

\ }

let g:ycm\_cache\_omnifunc = 1

let g:ycm\_use\_ultisnips\_completer = 1

"定义函数跟踪快捷健

nnoremap <leader>jd :YcmCompleter GoToDefinitionElseDeclaration<CR>