```python
import telnetlib,argparse,os,paramiko,tarfile


'''端口检测的主要实现过程'''
def getport(ip1,portname):
 # print(ip1,type(ip1),portname,type(portname))
 ip1=ip1.strip()
 portname=int(portname)
 try:
  telnetlib.Telnet(host=ip1, port=portname, timeout=2)
  print('>>>>>{0} success'.format(ip1))
  return True
  except Exception as e:
  print(e)
  return False

 '''sftp 发送文件的主要实现过程'''
 def sendfile(username,host,pwd,sf,ef):
  transport = paramiko.Transport((host, 22))
  transport.connect(username=username, password=pwd)
  sftp = paramiko.SFTPClient.from_transport(transport)
  jj=True
  try:
  # sftp.put('node-exporter.tar.gz', '/app/prometheus/exporter/node-exporter.tar.gz')
  # sftp.put('node_exporter.service','/usr/lib/systemd/system/node_exporter.service')
  sftp.put(sf,ef)
  except IOError as e:
  jj=False
  print('{0}原文件或者目的文件路径问题:'.format(sf),e)
  except Exception as e:
  jj=False
  print('{0}sft传输传输过程出现问题:'.format(ef),e)
  finally:
  sftp.close()
  return jj

 '''ssh密码登录的主要实现过程'''
 def exe_cmd(username,host,pwd,cmd,key=0):
```

```
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh.connect(hostname=host, port=22, username=username, password=pwd)
    stdin, stdout, stderr = ssh.exec_command(cmd)
    nameid = str(stderr.read()).replace('b', '').replace("'", '')

stdid=str(stdout.read()).replace('b','').replace("'",'').replace(r'\n','')
    if key:
    return stdid
    else:
    if nameid:
    # print('______{0}______faile'.format(host))
    return False
    else:
    # print('______{0}______命令执行成功'.format(host))
    return True



'''从文件中得到ip列表并去重的主要过程'''
def iplist(file):
    f=open(file)
    listda=[]
    for i in f:
    hh=i.strip().replace('\n','')
    if hh :
    listda.append(hh)
    print('___list____')
    print(listda,'\n',len(listda))
    list1=list(set(listda))
    print('___set___')
    print(list1,'\n',len(list1))
    return list1

'''从输入中得到ip列表'''
def iplist_line(strings):
    str_to_list=strings.strip().strip(':').split(':')
    list_to_set=set(str_to_list)
    getlist=list(list_to_set)
    return getlist

```

```
79
80
81  '''批量端口检测的主要过程'''
82  def port_test(iplist,port):
83   name=[]
84   print('\n\n')
85   for i in iplist:
86   print('host {0} start ceshi>>>>>>'.format(i))
87   getbool=getport(i,port)
88   print('主机{0}端口{1}开启结果:{2} '.format(i,port,getbool))
89   print('\n')
90   if getbool:
91   name.append(i)
92   print('port-test is complate')
93   print('print iplist:{0}'.format(name))
94   print('print NO iplist:
{0}'.format(list(set(iplist).difference(set(name)))))
95
96
97
98  '''批量文件分发接口无论文件夹或者文件通吃'''
99
100  def file1(list_name,username,pwd,sf,ef):
101   iplist11=[]
102   print('批量传输文件即将开始》》》》》》》》 ')
103   print('\n\n')
104   for i in list_name:
105   print('host {0} 传输文件开始'.format(i))
106   try:
107   if os.path.isfile(sf):
108   sendfile(username,i,pwd,sf,ef)
109   elif os.path.isdir(sf):
110   tarfilename=sf.split('/').pop()
111   sendfiletarname = '{0}.tar.gz'.format(tarfilename)
112   endfilepath=ef+'/'+sendfiletarname
113   cmd_name='tar xvf {0} -C {1};rm -rf {2}'.format(endfilepath,ef,endfilep
ath)
114
115   tar=tarfile.open(sendfiletarname,'w')
116   tar.add(sf)
117   tar.close()
```

```
118   sendfile(username, i, pwd,sendfiletarname,endfilepath )
119   exe_cmd(username,i,pwd,cmd=cmd_name)
120   iplist11.append(i)
121   print('{0}传输结果:true'.format(i))
122   except Exception as e:
123   print('{0}传输结果:false'.format(i))
124
125   return iplist11,list(set(list_name).difference(set(iplist11)))
126
127
128  '''定制化方法一'''
129  def fun1(filenames,username,pwd,list_name):
130   cmd1 = 'ls {0}|wc -l'.format(filenames)
131   cmd2 = 'mkdir -p /app/prometheus/exporter'
132   cmd3 = 'tar zxvf /app/prometheus/exporter/node-exporter.tar.gz -C /app/
prometheus/exporter;systemctl enable node_exporter;systemctl start node_exp
orter'
133   for i in list_name:
134   id = exe_cmd(username, i, pwd, cmd1, key=1)
135   print(id, 'id____')
136   if int(id):
137   print('检测得到{0}存在,将不再进行部署操作'.format(filenames))
138   else:
139   print('检测到{0}不存在,下面进行端口检测'.format(filenames))
140   if getport(i, 9100):
141   print('指定主机{0}端口9100被占用,将不再进行下面的操作,sorry!!'.format(i))
142   else:
143   print('下面将进行部署操作')
144   print('creat file {0} is {1}'.format('/app/prometheus/exporter', exe_cm
d(username, i, pwd, cmd2)))
145   sendfile(username, i, pwd, 'node-exporter.tar.gz', '/app/prometheus/exp
orter/node-exporter.tar.gz')
146   sendfile(username, i, pwd, 'node_exporter.service', '/usr/lib/systemd/s
ystem/node_exporter.service')
147   print('file 解压以及node-exporter 启动:{0}'.format(exe_cmd(username, i,
pwd, cmd3)))
148
149   print('{0} install node-expert complate'.format(i))
150   print('\n\n')
151
152   else:
153   pass
```

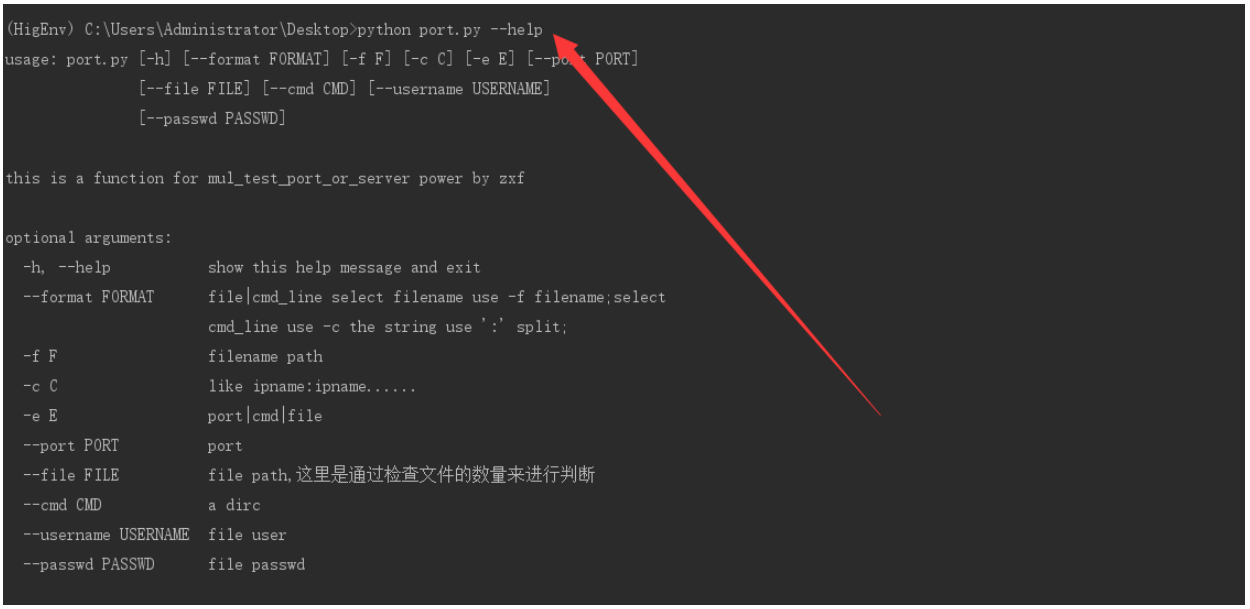
```
154
155 '''定制化命令一'''
156 def cmd1(list_name,all_cmd,username,pwd):
157  listmain = []
158  try:
159  list1 = eval(all_cmd)
160  listmain = list1.copy().copy()
161  del list1
162  except Exception:
163  print('输入的命令参数值错误，请重试')
164  exit(1)
165  for ip1 in list_name:
166  print('正在对{0}主机进行操作'.format(ip1))
167  for i in listmain:
168  bools = ''
169  if i['file']:
170  sf, ef = i['file'].strip().strip(':').split(':')
171  bools = sendfile(username, ip1, pwd, sf, ef)
172  elif i['cmd']:
173  bools = exe_cmd(username, ip1, pwd, i['cmd'])
174  elif i['port']:
175  bools =getport(ip1, i['port'])
176  if bools == i['source']:
177  continue
178  else:
179  exit(1)
180
181
182
183 '''参数的帮助信息的显示'''
184 def cmd_line():
185  parser= argparse.ArgumentParser()
186  parser.description = 'this is a function for mul_test_port_or_server po
wer by zxf'
187  parser.add_argument("--format", help="file|cmd_line select filename use
-f filename;select cmd_line use -c the string use ':' split;", type=str)
188  parser.add_argument("-f", help="filename path", type=str)
189  parser.add_argument("-c", help="like ipname:ipname......", type=str)
190  parser.add_argument("-e", help="port|cmd|file|fun",type=str)
191  parser.add_argument("--port", help="port ", type=str)
192  parser.add_argument("--files", help="sf:ef", type=str)
```

```
193  parser.add_argument("--filefun", help="file path,这里是通过检查文件的数量来进行判断", type=str)
194  parser.add_argument("--cmd", help="a dirc ", type=str)
195  parser.add_argument("--username", help="file user", type=str)
196  parser.add_argument("--passwd", help="file \n passwd", type=str)
197  args= parser.parse_args()
198  return args
199
200
201
202 '''在用户不用考虑方式的情况下获取ip列表'''
203 def all_get_ip_list(args):
204  if args.format=='file':
205  return iplist(args.f)
206  elif args.format=='cmd_line':
207  return iplist_line(args.c)
208
209
210 '''主方法的调度'''
211 def main(args,list_name):
212  if args.e=='port':
213  portname = args.port
214  port_test(list_name,portname)
215  elif args.e=='file':
216  filenames=args.files
217  username=args.username
218  pwd=args.passwd
219  sf,ef=filenames.strip().strip(':').split(':')
220  sf=sf.rstrip('/')
221  ef = ef.rstrip('/')
222  listip1,listip2=file1(list_name,username,pwd,sf,ef)
223  print('已经成功执行ip列表:',listip1)
224  print('未成功执行的ip列表:',listip2)
225  elif args.e=='cmd':
226  all_cmd=args.cmd
227  username = args.username
228  pwd = args.passwd
229  cmd1(list_name,all_cmd,username,pwd)
230  elif args.e=='fun':
231  print('install prometheus will start >>>>>>>>>>>>>>>>>')
232  filenames=args.filefun
```

```
233   username=args.username
234   pwd=args.passwd
235   fun1(filenames,username,pwd,list_name)
236
237
238 if __name__=='__main__':
239
240
241   args=cmd_line()
242   list_name =all_get_ip_list(args)
243   main(args,list_name)
244   # print(getport('10.141.90.220','22'))
245
```

例子:

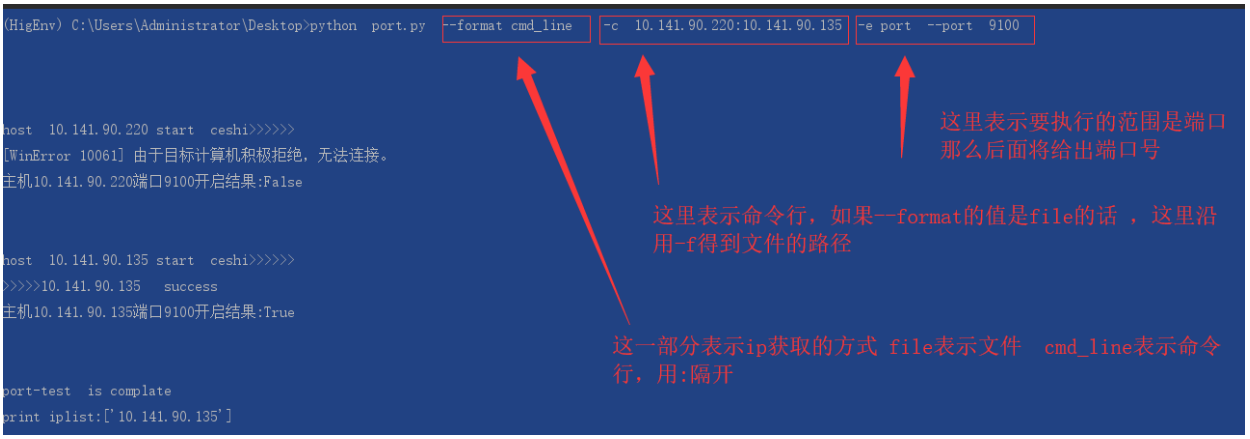
1/python port.py --help 显示帮助信息



2/端口检测

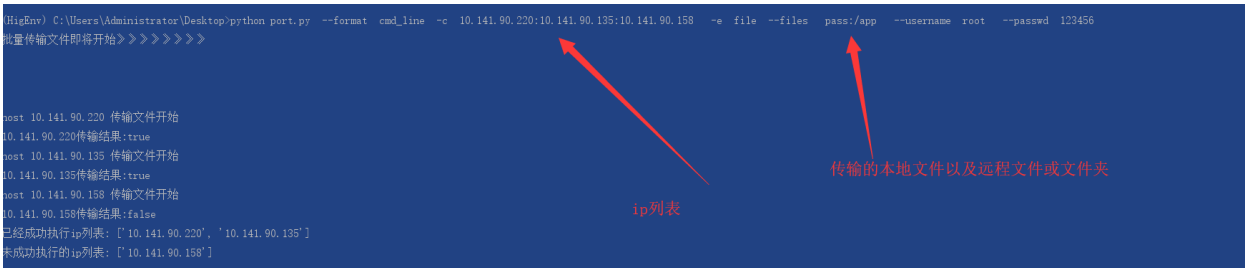
python port.py --format cmd_line -c 10.141.90.220:10.141.90.135 -e port --port 9100 从命令行获取ip检测端口

python port.py --format file -f iplist.text -e port --port 9100 通过文件获取ip列表进行端口检测



3/针对批量ip的传输文件

python port.py --format cmd_line -c 10.141.90.220:10.141.90.135:10.141.90.158 -e file --files pass:/app --username root --passwd 123456

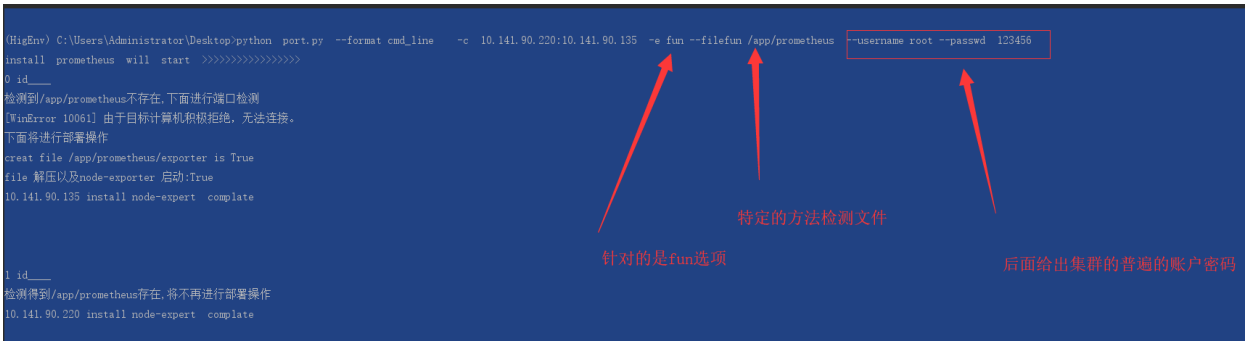


4/针对批量执行命令

python port.py --format file -f s.txt -e cmd --cmd "[{'port':'3306','file':'','cmd':'','source':False}, {'file':'pass.tar.gz:/root/pass.tar.gz','port':'','cmd':'','source':True}, {'cmd':'tar xvf /root/pass.tar.gz -C /root','file':'','port':'','source':True}]" --username root --passwd 123456

5/针对本次node-exporter安装所编写的方法

python port.py --format cmd_line -c 10.141.90.220:10.141.90.135 -e fun --filefun /app/prometheus --username root --passwd 123456 检测文件是否存在，然后执行操作


(HigEnv) C:\Users\Administrator\Desktop>python port.py --format cmd_line -c 10.141.90.220:10.141.90.135 -e fun --filefun /app/prometheus --username root --passwd 123456
install prometheus will start >>>>>>>>>>>>>>>>>>
0 id___
检测到/app/prometheus不存在,下面进行端口检测
[WinError 10061] 由于目标计算机积极拒绝，无法连接。
下面将进行部署操作
creat file /app/prometheus/exporter is True
file 解压以及node-exporter 启动:True
10.141.90.135 install node-expert complate
1 id___
检测得到/app/prometheus存在,将不再进行部署操作
10.141.90.220 install node-expert complate
针对的是fun选项
特定的方法检测文件
后面给出集群的普遍的账户密码