



# BLG 336E Analysis of Algorithms II

HOW TO USE CALICO

---

Istanbul Technical University

April 5th, 2021

The purpose of this document is to show to connect ITU's server  
and  
test your own source codes using a python module namely  
**Calico.**

Assignments are tested with different test cases using this tool.  
So when you test your source code before submitting it to the  
Ninova, we will have less issues such as “it is working on my local  
computer”. We will all be working on ITU’s server and on a  
Ubuntu system.

# How to connect ITU's ssh server?

Open terminal.

Type **ssh**.

```
(base) tardis$ssh
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
```

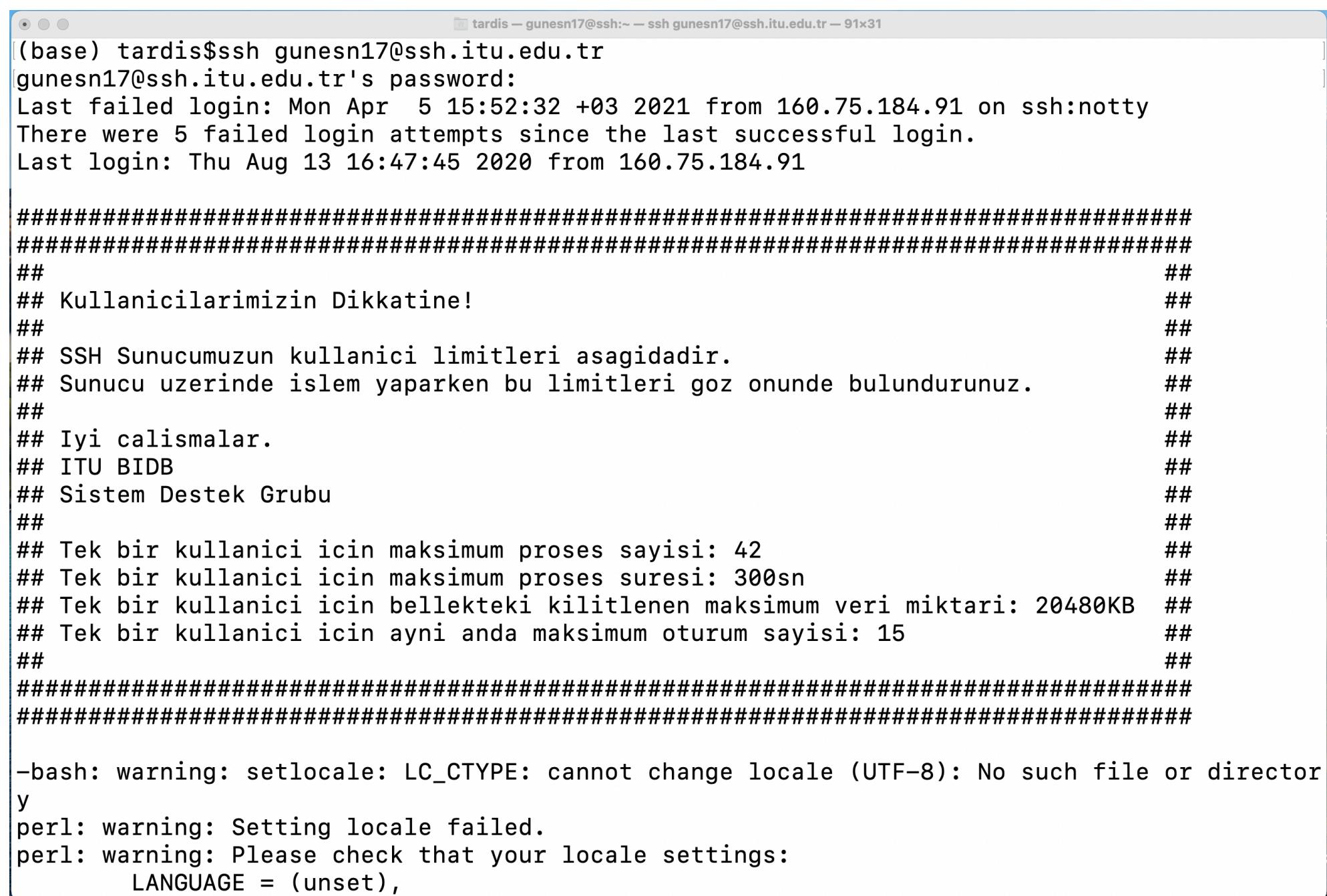
# How to connect ITU's ssh server?

Now enter name of the server you want to connect:

**itu\_user\_name@ssh.itu.edu.tr**

**EXAMPLE USAGE: ssh user@server\_name**

**ssh gunesn17@ssh.itu.edu.tr**



A screenshot of a terminal window titled "tardis — gunesn17@ssh:~ — ssh gunesn17@ssh.itu.edu.tr — 91x31". The window displays the following text:

```
(base) tardis$ssh gunesn17@ssh.itu.edu.tr
gunesn17@ssh.itu.edu.tr's password:
Last failed login: Mon Apr  5 15:52:32 +03 2021 from 160.75.184.91 on ssh:notty
There were 5 failed login attempts since the last successful login.
Last login: Thu Aug 13 16:47:45 2020 from 160.75.184.91

#####
## Kullanicilarimizin Dikkatine!
## SSH Sunucumuzun kullanici limitleri asagidadir.
## Sunucu üzerinde islem yaparken bu limitleri goz onunde bulundurunuz.
## Iyi calismalar.
## ITU BIDB
## Sistem Destek Grubu
## Tek bir kullanici icin maksimum proses sayisi: 42
## Tek bir kullanici icin maksimum proses suresi: 300sn
## Tek bir kullanici icin bellekteki kilitlenen maksimum veri miktari: 20480KB
## Tek bir kullanici icin ayni anda maksimum oturum sayisi: 15
## -bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
      LANGUAGE = (unset),
```



# One command: ssh

One argument: gunesn17@ssh.itu.edu.tr

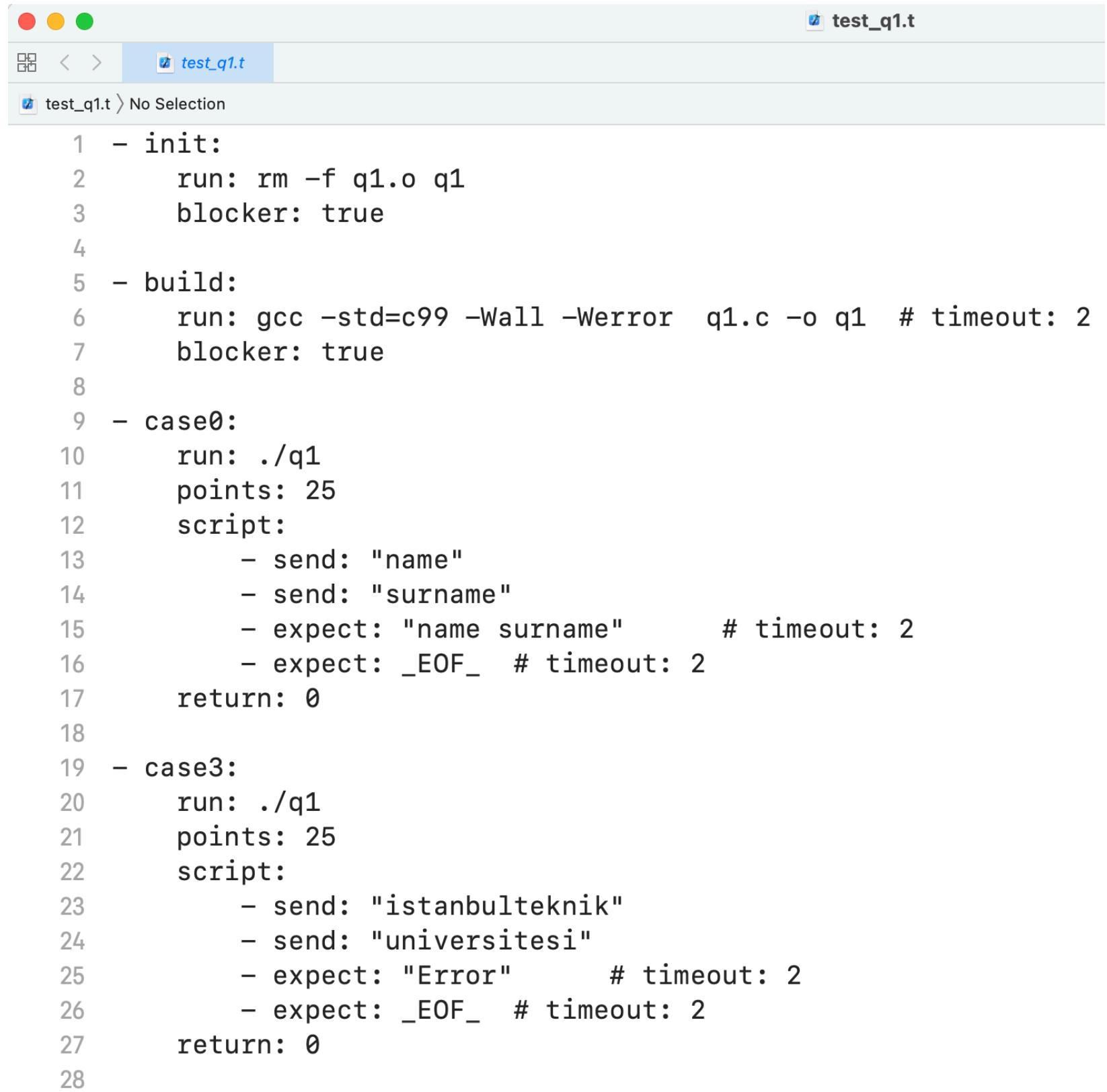
**ssh    gunesn17@ssh.itu.edu.tr**

Calico is a python module we use to evaluate  
programming assignments.

For more info: <https://pypi.org/project/calico/>

```
[gunesn17@ssh blg336e]$ python -m calico.cli --version
calico 1.1.2
[gunesn17@ssh blg336e]$
[gunesn17@ssh blg336e]$
[gunesn17@ssh blg336e]$ _
```

You need a test case file ending with .t

A screenshot of a Mac OS X-style text editor window titled "test\_q1.t". The file contains a Calico test script. The code uses YAML syntax to define test cases, including setup steps like removing files and blocking, building executables, and running them with specific inputs and expect statements.

```
1 - init:
2   run: rm -f q1.o q1
3   blocker: true
4
5 - build:
6   run: gcc -std=c99 -Wall -Werror q1.c -o q1 # timeout: 2
7   blocker: true
8
9 - case0:
10  run: ./q1
11  points: 25
12  script:
13    - send: "name"
14    - send: "surname"
15    - expect: "name surname"      # timeout: 2
16    - expect: _EOF_  # timeout: 2
17  return: 0
18
19 - case3:
20  run: ./q1
21  points: 25
22  script:
23    - send: "istanbulteknik"
24    - send: "universitesi"
25    - expect: "Error"      # timeout: 2
26    - expect: _EOF_  # timeout: 2
27  return: 0
28
```

All you need is:

```
python -m calico.cli test_case.t
```

```
[gunesn17@ssh blg336e]$  
[gunesn17@ssh blg336e]$  
[gunesn17@ssh blg336e]$ ls  
main.cpp  test_case.t  
[gunesn17@ssh blg336e]$  
[gunesn17@ssh blg336e]$  
[gunesn17@ssh blg336e]$ python -m calico.cli test_case.t  
init ..... PASSED  
build ..... PASSED  
case1 ..... 25 / 25  
Grade: 25 / 25  
[gunesn17@ssh blg336e]$  
[gunesn17@ssh blg336e]$  
[gunesn17@ssh blg336e]$
```

# How to use calico?

Here are the files, try it on your own:

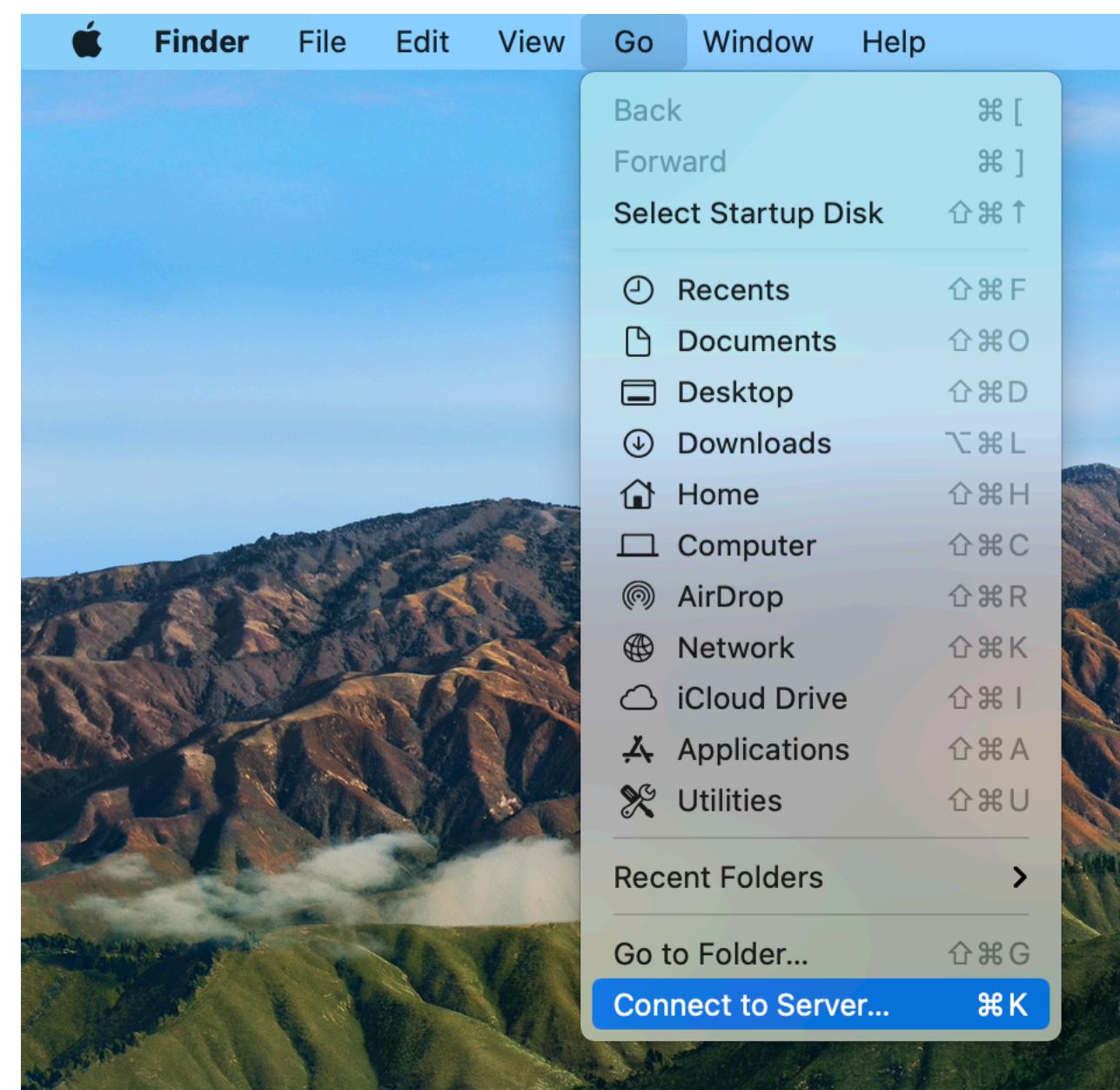
```
python -m calico.cli test_case.t
```

```
- init:  
    run: rm -f main.o main  
  
- build:  
    run: g++ -std=c++11 -Wall -Werror main.cpp -o main # timeout: 2  
  
- case1:  
    run: ./main  
    points: 25  
    script:  
        - expect: "Hello World"      # timeout: 2  
    return: 0
```

```
#include <iostream>  
  
using namespace std;  
  
int main(int argc, char **argv)  
{  
    cout << "Hello World" << endl;  
    return 0;  
}
```

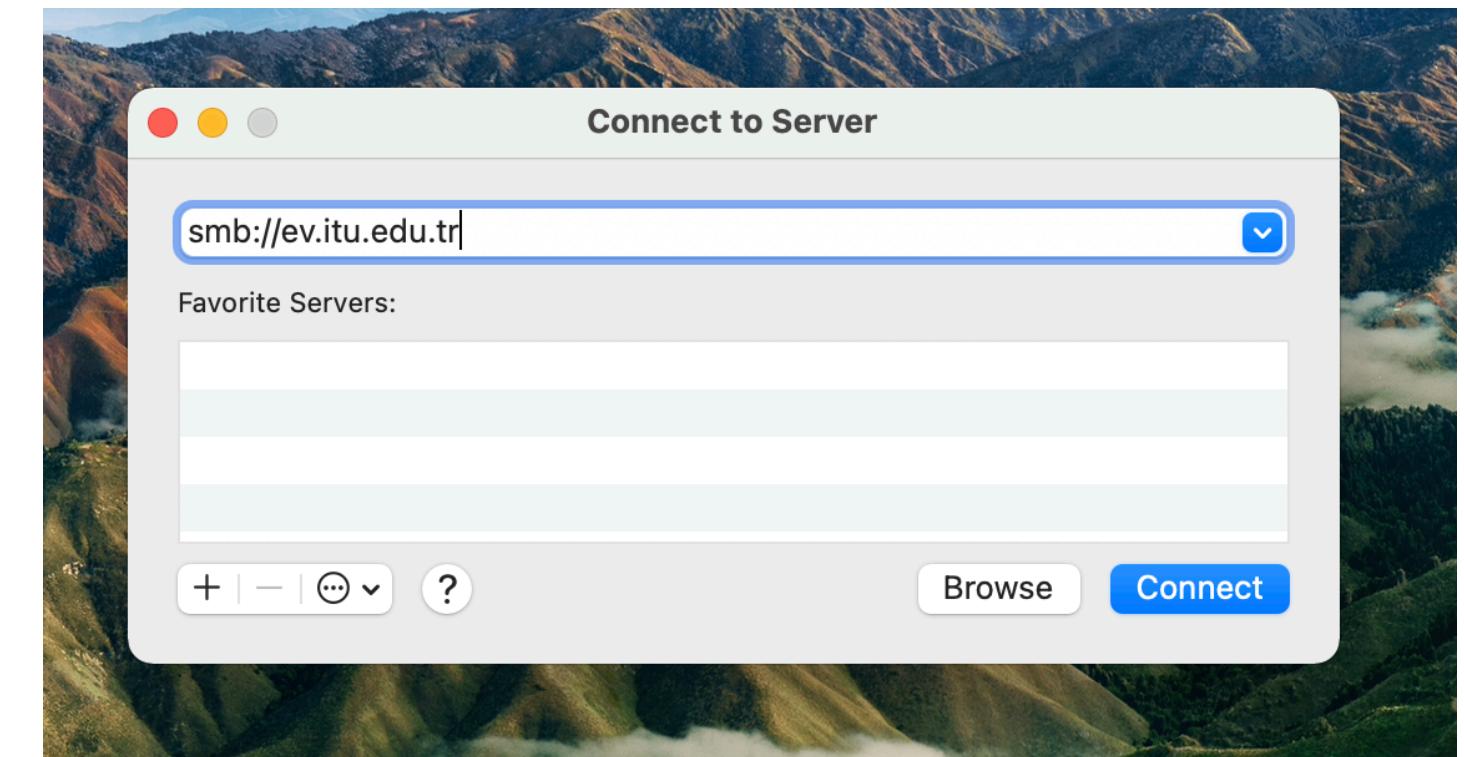
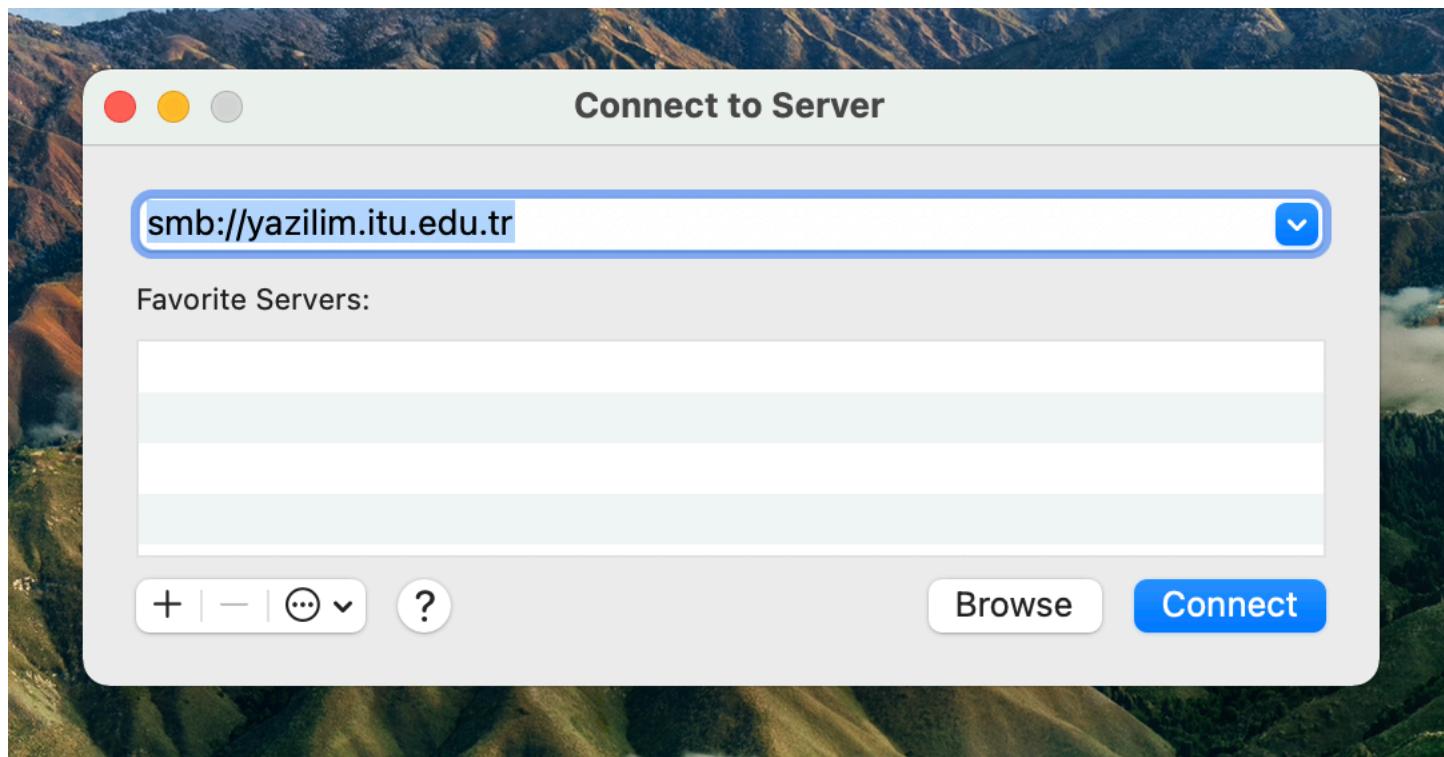
Instead of using terminal, you can connect to ssh server from finder.

For windows: <https://www.youtube.com/watch?v=DA-Uxz00QEU>



To connect **yazılım** server (**yazılım** is the name of the server), type: <smb://yazilik.itu.edu.tr>

To connect **ssh** server, type: <smb://ev.itu.edu.tr>



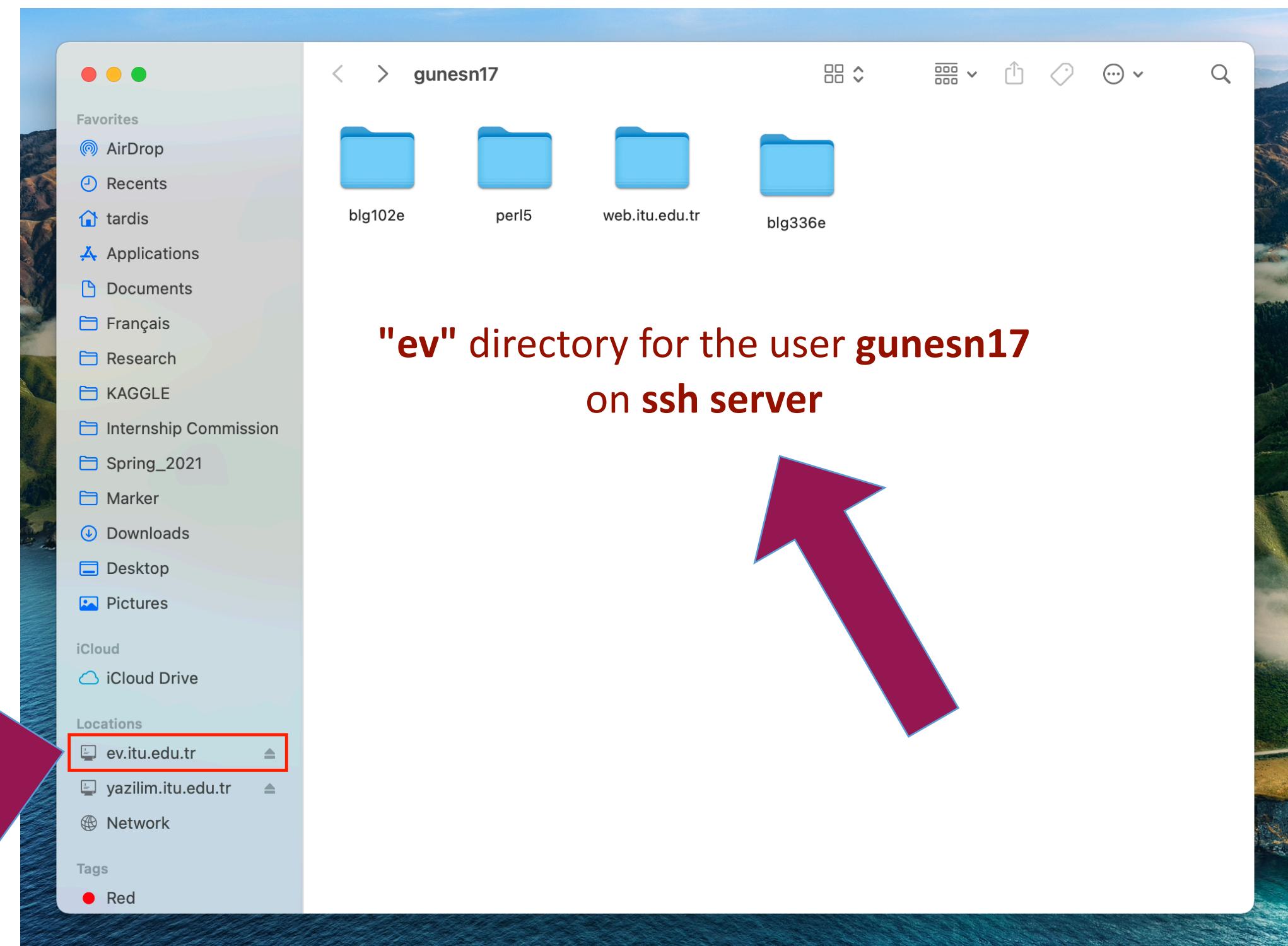
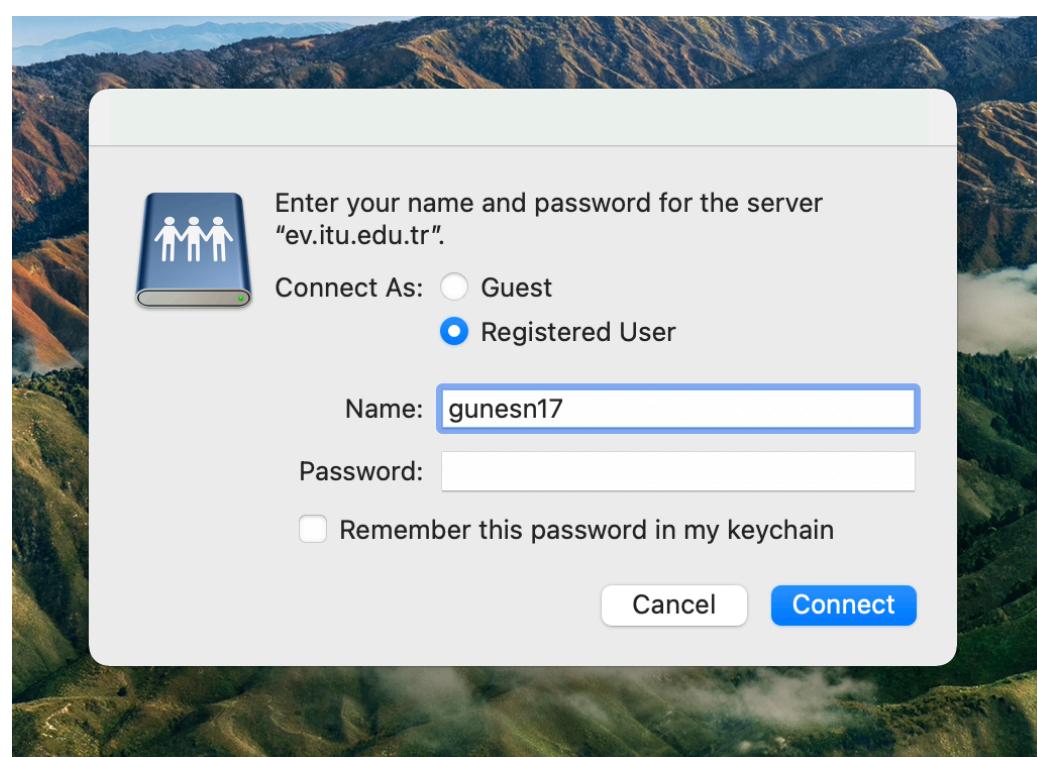
You are right, the name of the server is “**ev**”.

# Extras: How to connect to ssh server manually?

İTÜ



Write your password and log in. Now you can explore your files and you are not stuck with bash and nano.



"ev" directory for the user **gunesn17**  
on ssh server

If you can not connect to ssh server please first try to connect yazılım server following the instructions given in the link below.

Calico is already installed on ssh server.

<https://bidb.itu.edu.tr/bilgi-bankasi/yazilim-sunucusuna-erisim>

# Extras: How to download calico locally?

İTÜ



If you want to install calico module on your computer,  
you can use: **pip install calico**

For more information:

<https://pypi.org/project/calico/>

```
(default) tardis$ pip install calico
Collecting calico
  Using cached calico-1.2.0-py2.py3-none-any.whl (36 kB)
Collecting pexpect<5.0,>=4.6
  Using cached pexpect-4.8.0-py2.py3-none-any.whl (59 kB)
Collecting ruamel.yaml<0.16.0,>=0.15.41
  Downloading ruamel.yaml-0.15.100-cp36-cp36m-macosx_10_9_x86_64.whl (254 kB)
    |██████████| 254 kB 1.6 MB/s
Collecting ptyprocess>=0.5
  Using cached ptyprocess-0.7.0-py2.py3-none-any.whl (13 kB)
Installing collected packages: ptyprocess, ruamel.yaml, pexpect, calico
Successfully installed calico-1.2.0 pexpect-4.8.0 ptyprocess-0.7.0 ruamel.yaml-0
.15.100
(default) tardis$ calico --version
calico 1.2.0
(default) tardis$ calico
usage: calico [-h] [--version] [-d DIRECTORY] [--validate] [-q] [--log]
               [--debug] [-t TESTS [TESTS ...]] [--timeout TIMEOUT]
               spec
calico: error: the following arguments are required: spec
(default) tardis$
```

All you need is  
calico test\_case.t

```
(default) tardis$ls
main.cpp          test_case.t
(default) tardis$ 
(default) tardis$calico test_case.t
init ..... PASSED
build ..... PASSED
case1 ..... 25 / 25
Grade: 25 / 25
(default) tardis$ 
(default) tardis$ 
(default) tardis$
```

# Extras: What is it inside a test case file?

```
- init:  
    run: rm -f main.o main  
  
- build:  
    run: g++ -std=c++11 -Wall -Werror main.cpp -o main # timeout: 2  
  
- case1:  
    run: ./main  
    points: 25  
    script:  
        - expect: "Hello World"      # timeout: 2  
    return: 0
```

- Remove the files namely: main.o and main if exists.

```
rm -f main.o main
```

- Run the command given below:

```
g++ -std=c++11 -Wall -Werror main.cpp -o main
```

- Run the executable file with specified cases.

```
./main
```

# You are ready to go.



İTÜ

```
[gunesn17@ssh blg336e]$  
[gunesn17@ssh blg336e]$ exit  
logout  
Connection to ssh.itu.edu.tr closed.  
(base) tardis$  
(base) tardis$  
(base) tardis$  
(base) tardis$
```