

Laboratório 1

Objectivos:

- Desenvolvimento de aplicações Cliente/Servidor usando Sockets TCP/IP
- Criar máquinas virtuais na Google Cloud Platform
- Aceder remotamente a outro sistema através de cliente Secure Socket Shell (SSH)
- Medir tempos de execução incluindo latência no envio de mensagens entre processos locais e remotos

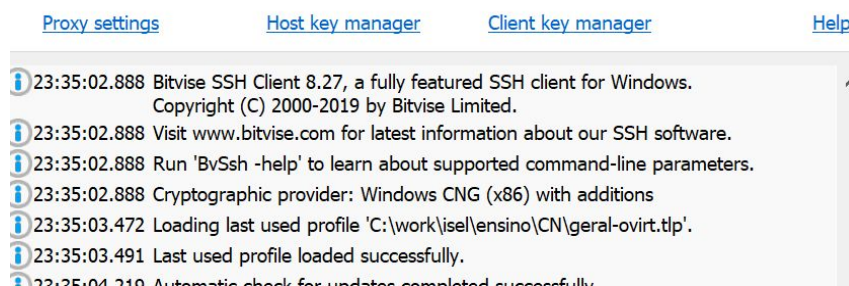
- 1) Considere os projetos IntelliJ, disponíveis no Moodle, que têm por base o cliente e servidor com *sockets* apresentado nas aulas. Neste exemplo o servidor recebe como argumentos uma carácter (*s* ou *c*) indicando se o atendimento de pedidos é sequencial ou em concorrência) e um porto onde fica à espera de pedidos. A aplicação cliente recebe como parâmetros o IP e o porto onde o servidor se encontra.

No projeto do servidor já está definido a criação de um artefato do tipo JAR executável (veja directoria `out\artifacts` após *build*).

- 2) Executando o servidor e várias instâncias do cliente na sua máquina, realize testes que permitam recolher os tempos de execução com o servidor em modo sequencial e em modo concorrente;
- 3) As máquinas virtuais criadas no GCP são acedidas via SSH com autenticação de chave pública e privada. O guião seguinte mostra como gerar um par de chaves pública/privada com o cliente SSH Bitvise em Windows:

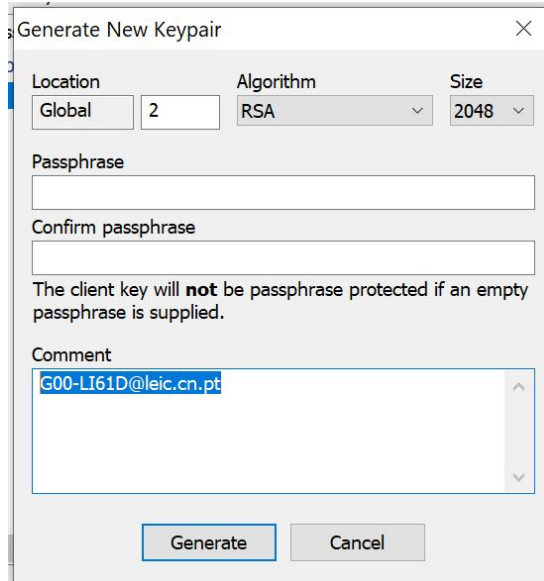
Para outros sistemas operativos, e outros clientes, sugerimos a consulta das instruções em <https://www.ssh.com/ssh/keygen/>, onde são usadas ferramentas de linha de comando para produzir o mesmo resultado.

- a) No cliente Bitvise aceda a “Client Key Manager”

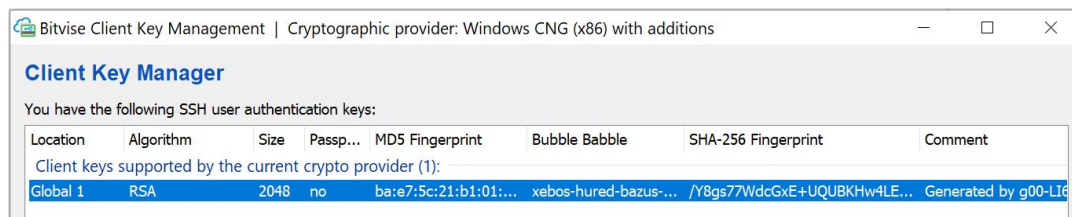


- b) Na zona inferior da janela, escolha “Generate New”
- c) Escolha uma password para proteger a chave privada, ou deixe em branco. **Na caixa de comentário** (“Comment”) indique um identificador com o formato `<nome>@cn.isel.pt`.

Sugere-se o nome do grupo como no projeto GCP, ex: G00-LI61D@cn.isel.pt.



- d) Selecione “Generate” para gerar o par de chaves e acrescentar à lista de chaves disponíveis no cliente Bitvise:



Location	Algorithm	Size	Passp...	MD5 Fingerprint	Bubble Babble	SHA-256 Fingerprint	Comment
Global 1	RSA	2048	no	ba:e7:5c:21:b1:01:...	xebos-hured-bazus-...	/Y8gs77WdcGxE+UQUBKHw4LE...	Generated by g00-LI6

- e) Exporte a chave pública escolhendo a opção “Export” da mesma janela. Indique o formato “OpenSSH” e exporte a chave pública para um ficheiro e diretoria à sua escolha.
- f) Visualize a chave pública exportada com um editor de texto (ex: code, notepad, ...).

- 4) Usando a conta GCP do grupo, no serviço Compute Engine crie 1 instância de máquina virtual do tipo ‘f1.micro’, selecionando a opção correspondente no menu “Machine type”:

- a) Ative HTTP e HTTPS na firewall.
- b) Click em “Management, security, disks, networking, sole tenancy” e depois no tab “Security”. Copie a chave pública SSH gerada no ponto 1 para o formulário disponível. Note que o formato imposto pelo formulário é: <protocol> <key-blob> <username@example.com>, o qual corresponde ao formato da chave gerada no ponto 1.c. Atenção ao fazer *copy/paste* a partir do ficheiro, onde guardou a chave no ponto 1.e, verificando que a última linha não tem um <Enter>.

Add tags and firewall rules to allow specific network traffic from the Internet.

- ☒ Allow HTTP traffic
- ☒ Allow HTTPS traffic

Management... **Networking** Sole Tenancy

Shielded VM ?
Select a shielded image to use shielded VM features.
Turn on all settings for the most secure configuration.

- ☐ Turn on Secure Boot ?
- ☐ Turn on vTPM ?
- ☐ Turn on Integrity Monitoring ?

SSH Keys
These keys allow access only to this instance, unlike [project-wide SSH keys](#) [Learn more](#)

☐ Block project-wide SSH keys
When ticked, project-wide SSH keys cannot access this instance. [Learn more](#).

G00-LI61D

```
MBH3s8Taz4J41zg747AXbdRuqU6UNVFZyEtGcZ6+A  
jZCp6U1ExU+1cpxLKa7bv1jrDVm/SRmKcH30a/2BQ  
KZfPRh2yeWM54TYitov0gTg5NYXyTc6oE/RGi0wq6  
yvvWJ6GgL+4BddnWZ0kiW2KCpvW+qRheAXj//3zR9  
jbBJpus5khicxYR06I3l8EruybX8HwZkEMt0/d8V+  
rgjcM1Xcv5f3EyWtCu1pTc8PWetGDL79jgy4NvAc  
cSofJ4QnJvB G00-LI61D
```

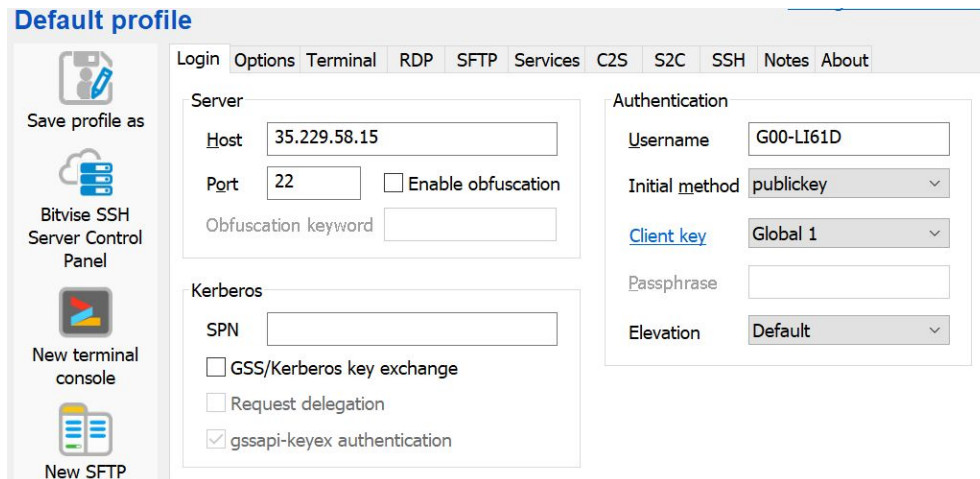
- c) Crie a VM e verifique na consola Web do GCP que a máquina foi iniciada e tem um IP externo:

VM instances [+ CREATE INSTANCE](#) [IMPORT VM](#) [REFRESH](#) [▶](#) [■](#) [🔄](#) [🗑️](#)

[Columns](#)

<input type="checkbox"/> Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect
<input type="checkbox"/> <input checked="" type="checkbox"/> instance-1	us-east1-b			10.142.0.2 (nic0)	35.229.58.15 🔗	SSH ▼ ⋮

- d) Aceda à VM através do cliente SSH. O utilizador é o indicado no ponto 1.c), ex: G00-LI61D, o método inicial é “public key” e a “Client key” tem de indicar a entrada criada anteriormente no ponto 1.d).



- e) Após login, verifique o correto acesso à VM. Não se esqueça de desligar a VM quando não a estiver a usar, usando o botão “Stop” na consola Web do GCP.

```
G00-LI61D@35.229.58.15:22 - Bitvise xterm - G00-LI61D@instance-2: ~
G00-LI61D@instance-2:~$ cat .ssh/authorized_keys
# Added by Google
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDcuct4o7LCPfEsFWg3puSmHojqBeYk00YtIyJBojfcF5mFIYVe
RdOHsoJmJWz1Uu1AhQImZor21Y6j2YJmmqOMBH3s8Taz4J41zg747AXbdRuqU6UNVFZyEtGcZ6+AjZCp6U1ExU+1
cpxLKa7bv1jrDvm/SRmKcH30a/2BQKZfPRh2yewM54TYitov0gTg5NYXyTc6oE/RGi0wq6ywwWJ6GgL+4BddnWZ0
kiW2KcPvW+qRheAXj//3zR9jbbJpus5khicxYR06I3l8ErubyX8HwZkEMt0/d8V+rgjcM1Xcv5f3EyWtCu1pTc8P
WeTGDl79jgy4NvAcicSofJ4QnJvB G00-LI61D
G00-LI61D@instance-2:~$
```

- 5) Instale o JDK 8 usando o comando “`sudo apt install openjdk-8-jdk`”
- 6) Faça *upload* do JAR do servidor do projeto do ponto (1) para a sua VM na GCP. Execute-o e repita os testes que realizou no ponto (2), executando o cliente no seu computador e o servidor na VM GCP que criou.