

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

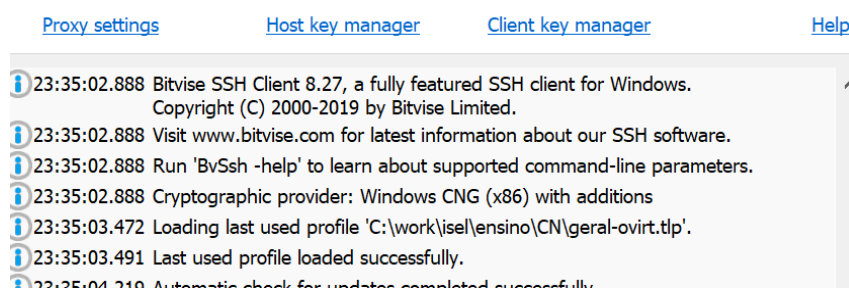
- 1) Considere os projetos IntelliJ, disponíveis no Moodle, que têm por base o cliente e o servidor com *sockets* apresentados nas aulas. Neste exemplo o servidor recebe como argumentos um carácter (**s** ou **c**) indicando se o atendimento de pedidos é sequencial ou em concorrência, e um porto onde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

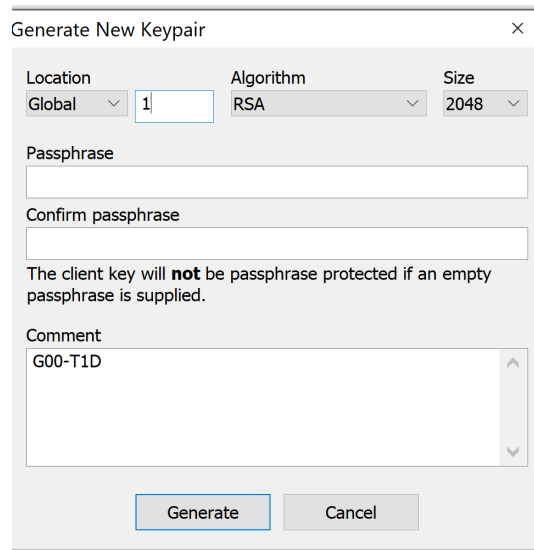
- 2) Executando o servidor e várias instâncias do cliente na sua máquina, realize testes que permitam recolher os tempos de execução com o servidor em modo sequencial e em modo concorrente;
- 3) As máquinas virtuais criadas no GCP são acedidas via SSH com autenticação de chave pública e privada. O guião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

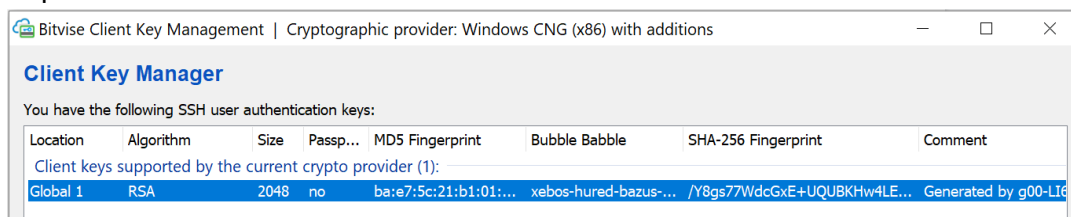
- a) No cliente Bitvise aceda a “Client Key Manager”



- 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 <grupo>-<turma>. Use o nome do grupo e turma como no projeto GCP, por exemplo, G00-T1D.

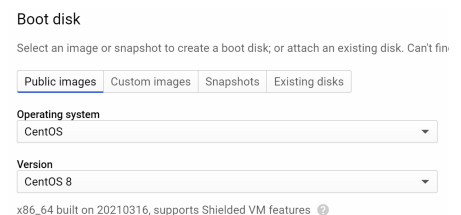


- d) Selecione “Generate” para gerar o par de chaves e acrescentar à lista de chaves disponíveis no cliente Bitvise:



- 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: VS Code, Notepad, ...).

- 4) Usando a conta GCP do grupo de alunos, no serviço Compute Engine crie 1 instância de máquina virtual selecionando (Series N1 Machine Type ‘f1.micro’) e sistema operativo (Boot Disk) CentOS version 8



- 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: `ssh-rsa <key-blob> G00-T1D`, o qual corresponde ao formato da chave gerada no ponto 3.c. Atenção ao fazer *copy/paste* a partir do ficheiro, onde guardou a chave no ponto 3.e, verificando que a última linha não tem um <Enter>.

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

☐ Allow HTTP traffic
☐ Allow HTTPS traffic

Management **Security** Disks Networking Sole Tenancy

Shielded VM ?
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 checked, project-wide SSH keys cannot access this instance [Learn more](#)

G00-T1D

```
dW+y04JuzWg2RZCvRvA57oDja+6axai1JUyFHuQm8  
OW2eygkrHisqfJWRR2DfXbQLmsZR5TigEsAke0q1  
CkQgv66F32QsojIkmkg5WeXfE5EUigYsJY3UNqkK1  
YjUZczD1j4raBI/8M1HU4z2W7tszJe0n+U1TeeXM2  
ijVYMZ8zdcBr0FJ3kkmD//IUrRQBUYfwidQ07J72+  
Wmx04K7Yb3NYtW76joi8St5Q+L9/eQ0XNahor1v0h  
ejQfgEMX74R G00-T1D
```

+ Add item

- 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

▶

■

🔄

🗑

☰

Filter VM instances

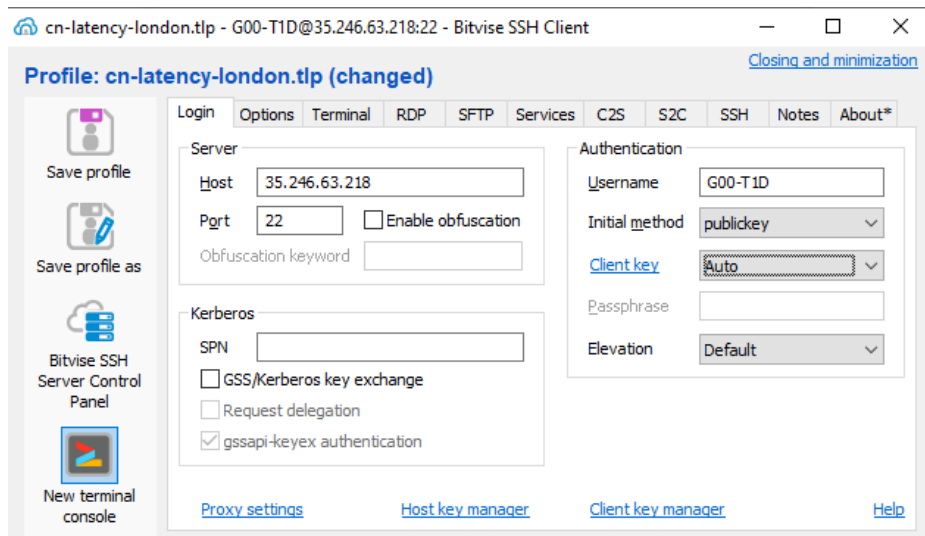
?

Columns

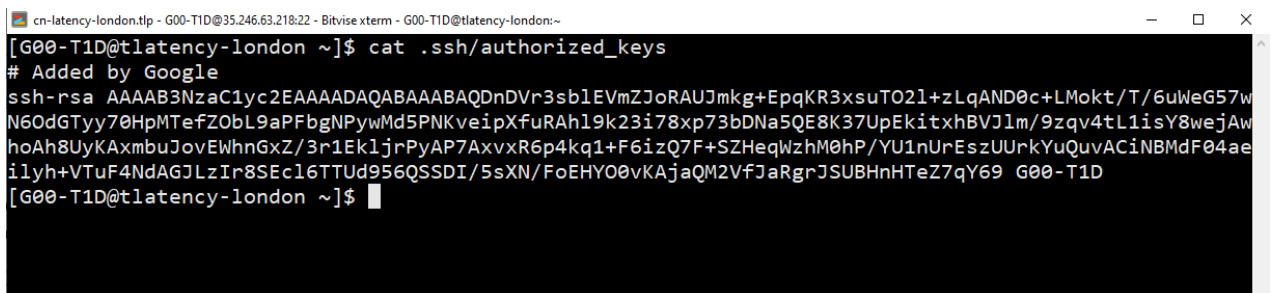
▼

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

- d) Aceda à VM através do cliente SSH (ver figura seguinte). O utilizador é o indicado no ponto 3.c), ex: G00-T1D, o método inicial é “public key” e a “Client key” tem de indicar a entrada criada anteriormente no ponto 3.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.



- 5) Instale o JDK 11 usando o comando “`sudo yum install java-11-openjdk-devel`”
- 6) Faça *upload* do JAR do servidor baseado em *sockets* do projeto do ponto (1) para a sua VM na GCP. Execute-o na VM e repita os testes que realizou no ponto (2), executando o cliente no seu computador. Note que o cliente Bitvise tem a opção de fazer “Secure Copy”:

