

## Laboratório 1

### Objectivos:

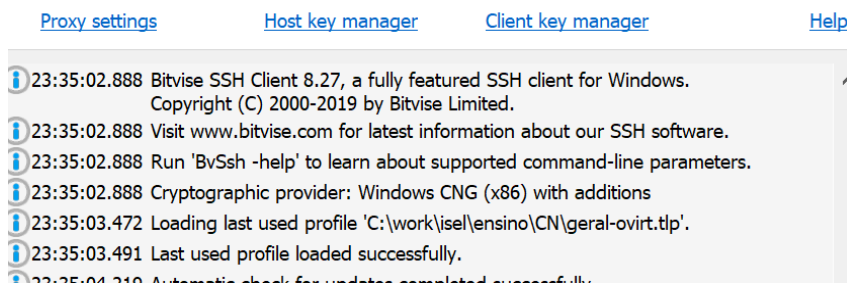
- Criar conta e projeto na Google Cloud Platform (GCP)
- Criar máquinas virtuais (VM) na Google Cloud Platform
- Aceder remotamente a uma VM através de cliente Secure Socket Shell (SSH)
- Execução de um artefacto Java numa VM acessível via HTTP em diferentes portos TCP/IP usando *Firewall rules*.

- 1) Seguindo o guião “*CN-Registo na Google Cloud Platform como Aluno - Verao2324.pdf*” garanta que acede à sua conta e projeto na consola web do GCP.
- 2) As máquinas virtuais que vão ser criadas no GCP são acedidas via SSH com autenticação de chave pública e privada. A aplicação cliente SSH que se recomenda para o Windows é o Bitvise (<https://www.bitvise.com/ssh-client-download>). Outros sistemas operativos têm soluções semelhantes.

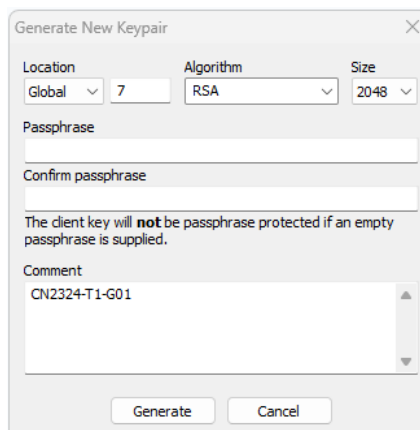
As alíneas seguintes mostram como cada aluno pode gerar um par de chaves pública/privada com o cliente SSH Bitvise em Windows:

*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”

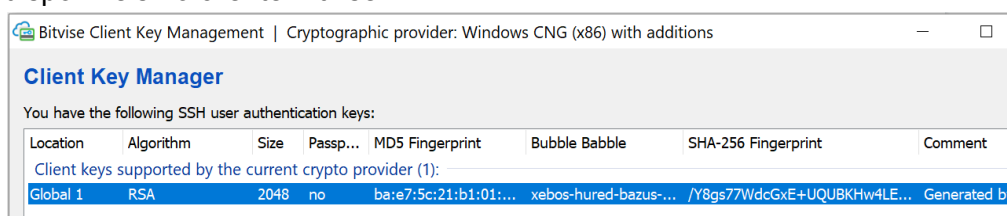


- b) Na zona inferior da janela, escolha “Generate New”
- c) Escolha uma password para proteger a chave privada, ou deixe em branco. **Na caixa de comentário** (“*Comment*”) indique um identificador com o formato CN2324-<turma>-<grupo>. Use o nome do grupo e turma como no projeto GCP, por exemplo, CN2324-T1-G01. Note que cada aluno deverá ter um par de chaves diferentes mas usar o mesmo identificador, o qual será o nome de utilizador a usar na sessão SSH para a VM.



Generate New Keypair dialog box. Fields: Location (Global), Algorithm (RSA), Size (2048), Passphrase, Confirm passphrase, Comment (CN2324-T1-G01). Buttons: Generate, Cancel.

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



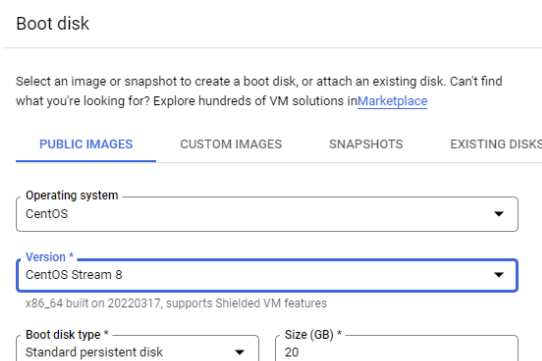
Bitvise Client Key Management window. Table of SSH user authentication keys:

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

- 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, ...). O formato da chave deve ter três partes: `ssh-rsa <chave> <identificador do grupo>`. Caso falte a última parte, complete com o identificador.

- 3) Usando a conta GCP do grupo de alunos, no serviço *Compute Engine* crie 1 instância de máquina virtual selecionando (*Series E2 Machine Type ‘e2-small’*) e sistema operativo (*Boot Disk*) CentOS Stream 8.

- a) Para permitir ligações ao porto 80 e 443 da VM, ative as opções HTTP e HTTPS na *firewall*.
- b) Clique em “*Advanced Options*” e depois selecione “*Security*” e em seguida “*VM Access*”. Adicione um item “*Add item*” na opção de “*Add manually generated SSH keys*”.



Boot disk configuration interface. Fields: Operating system (CentOS), Version \* (CentOS Stream 8), Boot disk type \* (Standard persistent disk), Size (GB) \* (20).

Copie integralmente para a caixa de texto disponível a chave pública SSH gerada e exportada anteriormente. Embora possa posteriormente adicionar as chaves dos restantes alunos do grupo, pode já nesta fase adicionar as várias chaves.

#### Add manually generated SSH keys

Add your own keys for VM access through a 3rd-party tool. You cannot use these keys when IAM-based access (using OS Login) is enabled. [Learn more](#)

SSH key 1 \*

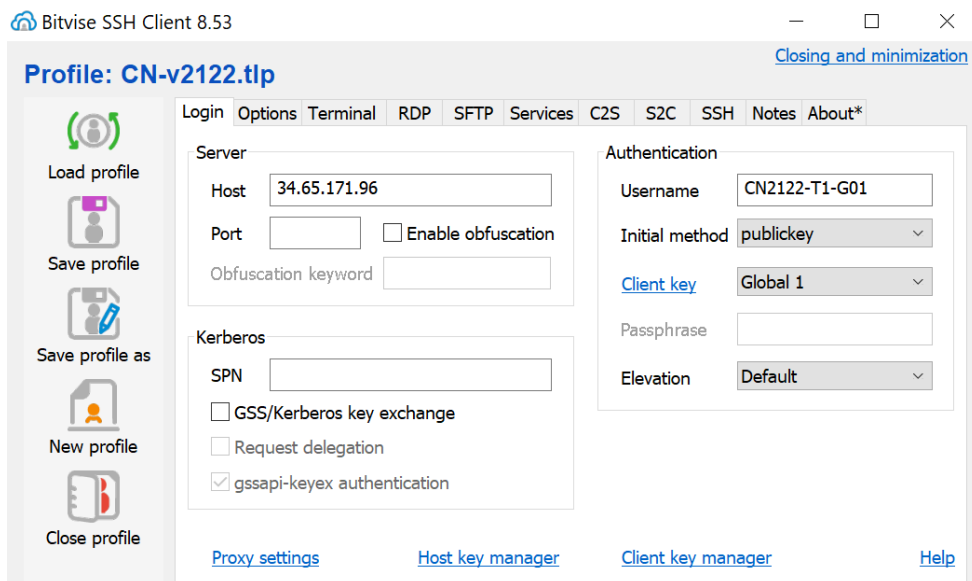
  
Enter public SSH key  
[+ ADD ITEM](#)

- c) Crie a VM e verifique na consola Web do GCP que a máquina foi iniciada e tem um IP externo (endereço IP público) acessível em qualquer localização:

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"/>	✓ instance-1	us-east1-b			10.142.0.2 (nic0)	35.229.58.15 ↗	SSH ▾ ⋮

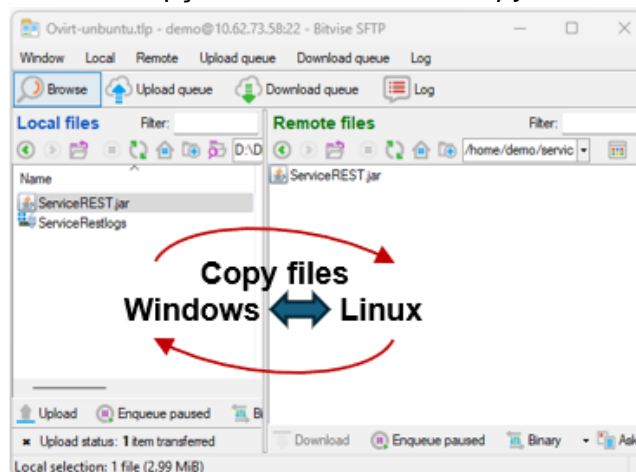
- d) Aceda à VM através do cliente SSH (ver figura seguinte). O utilizador é o indicado anteriormente (ex: CN2324-T1-G01) o método inicial é “public key” e a “Client key” tem de indicar a entrada correta (ex: *Global 1*).



- 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. Para ver a chave instalada na VM pode executar o seguinte comando Linux: `cat .ssh/authorized_keys`

- 4) Instale o *runtime* Java JDK 11, usando o comando com permissões de *super user* “`sudo yum install java-11-openjdk-devel`”
- 5) Em anexo ao enunciado existe o artefacto `java ServiceREST.jar`, que implementa um serviço HTTP REST, por omissão no porto 80, mas que ao ser lançado suporta a passagem como argumento outro porto TCP/IP (`java -jar ServiceREST.jar [porto]`). O serviço disponibiliza as seguintes rotas:  
`http://<host ip>[:porto]/ping`  
`http://<host ip>[:porto]/hello/<some name>`  
`http://<host ip>[:porto]/calc/number{+,-,*}number`  
Ex: `http://<host ip>[:porto]/calc/5*3` retorna 15
- 6) Execute o serviço localmente na sua máquina com diferentes portos e usando um browser HTTP aceda ao serviço experimentando as diferentes rotas, indicando como *<host ip>* o endereço *localhost*.
- 7) Faça *upload* do JAR (`ServiceREST.jar`) para a sua VM na GCP. Execute na VM o serviço com o porto por omissão 80, com o seguinte comando: `sudo java -jar ServiceREST.jar`. (Note que precisa de executar como *super user* por restrições do sistema operativo Linux na utilização do porto 80).

Note que o cliente Bitvise tem a opção de fazer “*Secure Copy*”:



- 8) Usando um browser HTTP na sua máquina local, aceda ao serviço em execução na VM, experimentando as diferentes rotas com o *<host ip>* no endereço público da VM no GCP.
- 9) Execute outra instância do serviço na VM do GCP, usando o porto 7500:  
(`java -jar ServiceREST.jar 7500`)
- 10) Na sua máquina local execute a rota `http://<VM ip>:7500/ping` e verifique que não consegue obter resposta.
- 11) No seu projeto GCP crie um *firewall rule* para permitir acessos ao porto 7500 e verifique posteriormente que já consegue aceder ao serviço a partir de qualquer computador.