

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.

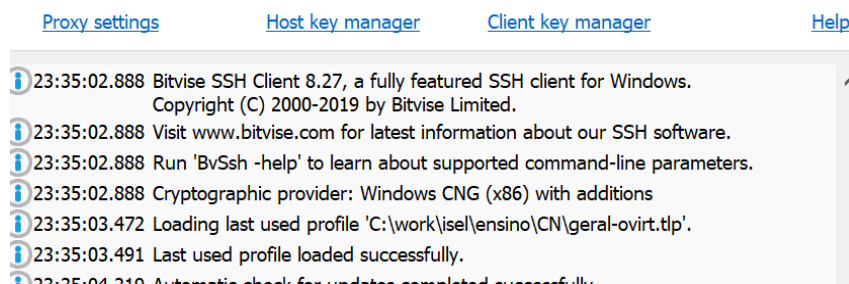
No projeto do servidor defina a criação de um artefato do tipo JAR executável (no menu *Project Structure - Artifacts - Add Artifact* (clique em **+**) - JAR - *From modules with dependencies* - Selecione *Main Class* - OK). Após *Build Artifacts*, verifique na directoria `out\artifacts` a existência de um JAR executável (`java -jar Server.jar {s|c} <TCP port>`).

- 2) Executando o servidor e várias instâncias do cliente na sua máquina, realize testes que permitam recolher e tirar conclusões sobre 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. 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.

O guião seguinte mostra como gerar um par de chaves pública/privada com o cliente SSH Bitvise em Windows:

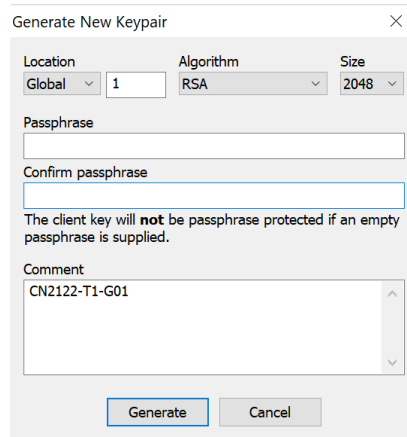
Para outros sistemas operativos, e outros clientes, sugerimos a consulta das instruções em <https://www.ssh.com/ssh/keygen/>, onde são usadas ferramentas de linha de comando para produzir o mesmo resultado.

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

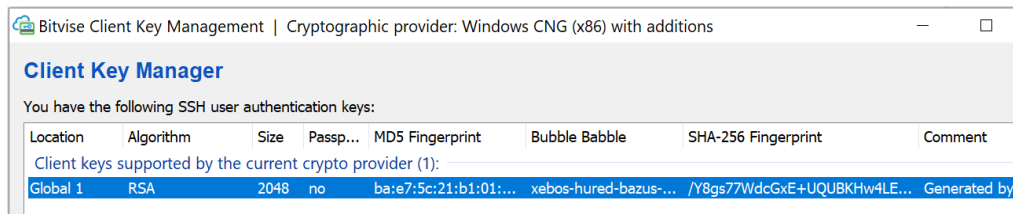


- b) Na zona inferior da janela, escolha “Generate New”

- c) Escolha uma password para proteger a chave privada, ou deixe em branco. **Na caixa de comentário** (“Comment”) indique um identificador com o formato CN2122-<grupo>-<turma>. Use o nome do grupo e turma como no projeto GCP, por exemplo, CN2122-T1-G01.

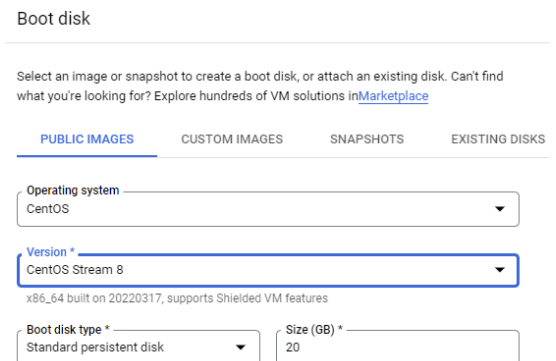


- d) Selecione “Generate” para gerar o par de chaves e acrescentar à lista de chaves disponíveis no cliente Bitvise:



- 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 E2 Machine Type ‘e2-micro’) e sistema operativo (Boot Disk) CentOS Stream 8.



- a) Ative HTTP e HTTPS na firewall.
- b) Clique em “Networking, Disks, Security, Management...” e depois selecione “Security” e em seguida “Manage Access”. Adicione um item na opção de “Add manually generated SSH keys”.

Copie integralmente para o formulário disponível a chave pública SSH gerada e exportada anteriormente.

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](#)

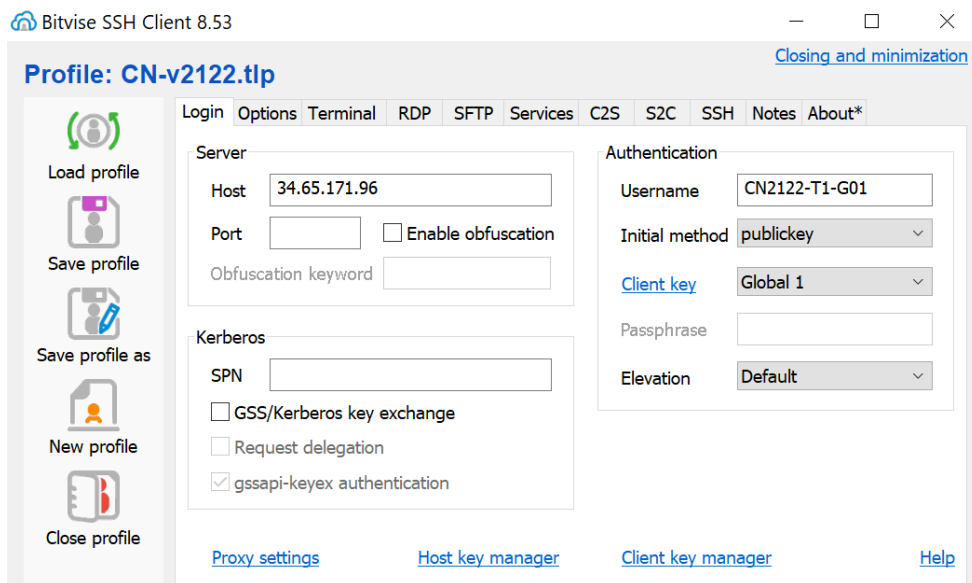
- c) Crie a VM e verifique na consola Web do GCP que a máquina foi iniciada e tem um IP externo:

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"/>	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: CN2122-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`

- 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 Bitwise tem a opção de fazer “Secure Copy”:

