

Computação na nuvem

ISEL – LEIRT / LEIC / LEIM

- Introdução à virtualização
- Configuração de VMs no serviço Compute Engine da Google

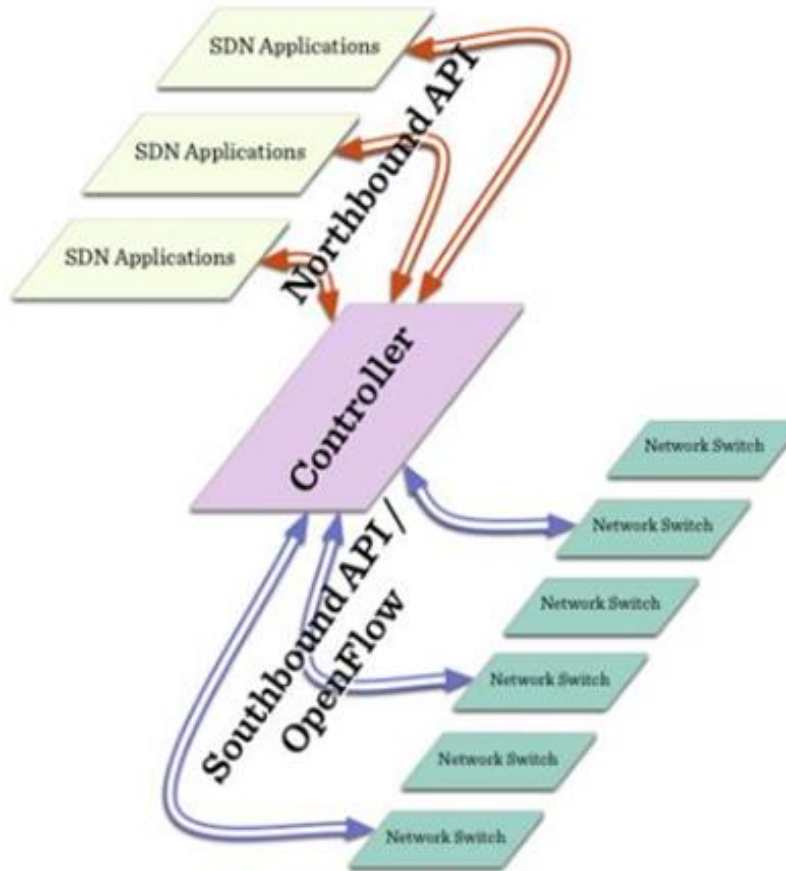
José Simão jsimao@cc.isel.ipl.pt ; jose.simao@isel.pt

Luís Assunção lass@isel.ipl.pt ; luis.assuncao@isel.pt

Tipos de virtualização

- A virtualização pode ser aplicada a vários tipos de componentes:
 - *Software Defined Networks (SDN)*
 - *Desktops* virtuais e remotos
 - *Software Defined Storage* (como estudamos no *Google Storage*)
 - Virtualização de ambientes de execução de software, nomeadamente na execução de aplicações desenvolvidas em Java

Virtualização de rede



SDN Applications: Usam uma API para definir os requisitos de rede através de um controlador SDN, consumindo uma rede virtual;

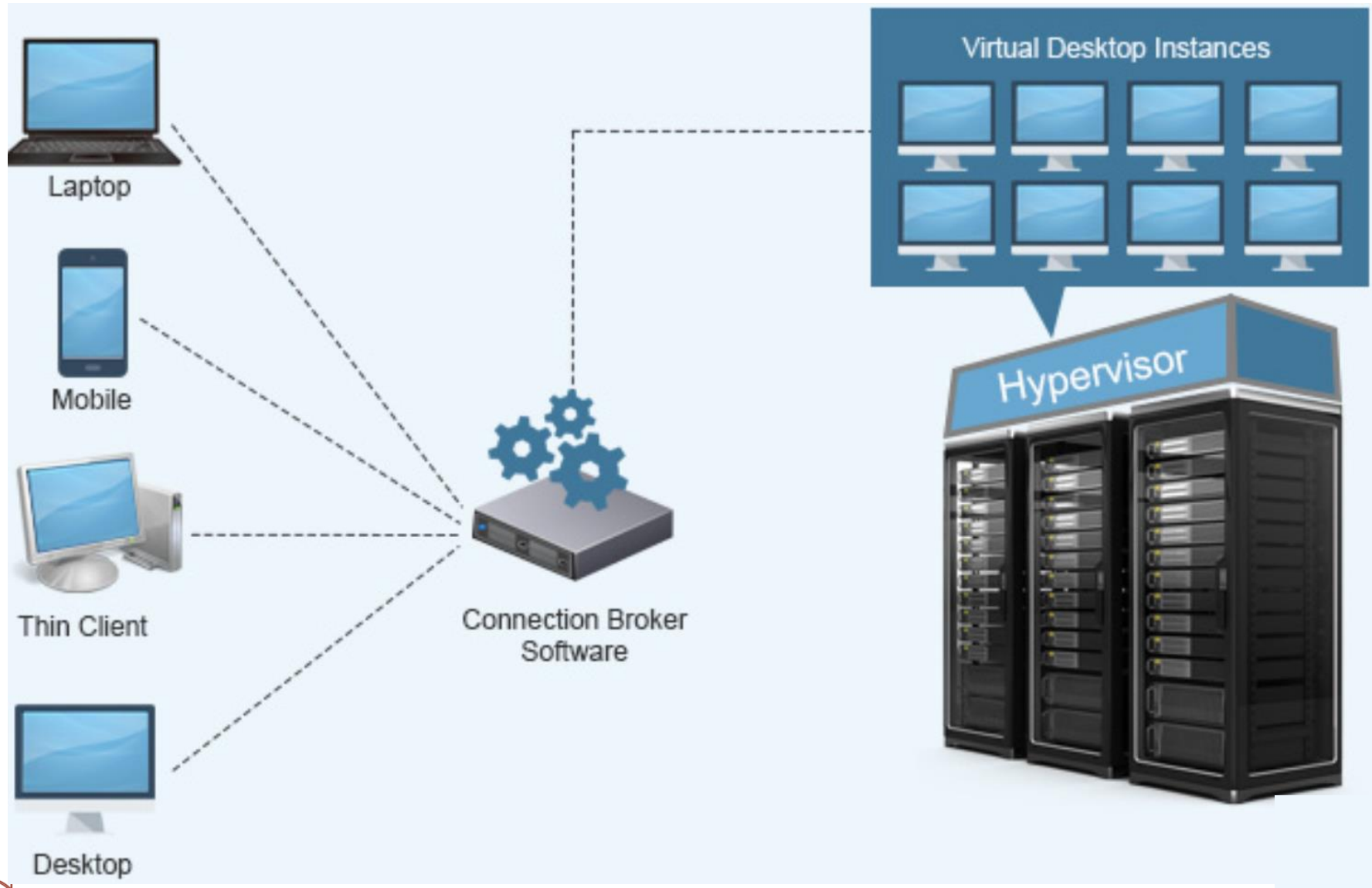
SDN Controller: Proporciona uma rede virtual às aplicações, traduzindo os requisitos de rede para os dispositivos físicos de acesso à rede física, através de protocolos standard (ex. *OpenFlow*)

Open Flow: Desenvolvido pela *Stanford University*, e gerido pela *Open Networking Foundation (ONF)* tem por objetivo, criar uma linguagem de programação de *switches* para controlo de tráfego (*flows*) e notificação de eventos na rede)

Virtual Desktops - Virtual Desktop Infrastructure (VDI)

- Executar sistemas operativos e aplicações desktop em máquinas virtuais alojadas em servidores remotos;
- Os utilizadores acedem ao desktop e aplicações a partir de qualquer computador pessoal ou browser através de um protocolo (ex: RDP, SPICE), como se tudo estivesse na máquina local;
- A diferença é que as aplicações são geridas centralmente, reduzindo drasticamente os custos de gestão/administração, nomeadamente adicionar ou substituir as aplicações por novas versões. A segurança é mais simples e torna o processo de *backup* de dados mais simples e rápido;

Virtual Desktop Infrastructure (VDI)



Porquê a virtualização?

- Consolidação e sistemas legados
 - Consolidação de vários servidores/serviços no conjunto mínimo de hardware para minimizar custos com a infraestrutura (menos espaço para mais serviços; menos equipamento de rede, menos energia, recursos humanos, ...), melhorando o nível de utilização do hardware disponível através da partilha de recursos
 - Sistemas legados (*legacy applications*) podem executar-se numa VM em infraestruturas físicas mais recentes e eficazes
- Isolamento
 - Falhas de segurança que exploram o sistema operativo, ou erros em componentes de software, ficam confinadas à VM
 - VMs podem ser usadas para criar ambientes isolados mais seguros (*sandboxes*) para executar aplicações suspeitas ou testar falhas
- Ambientes de desenvolvimento/investigação
 - Facilidade de criação de ambientes controlados com *stack* de software bem definido, ou redes de servidores com diferentes sistemas operativos

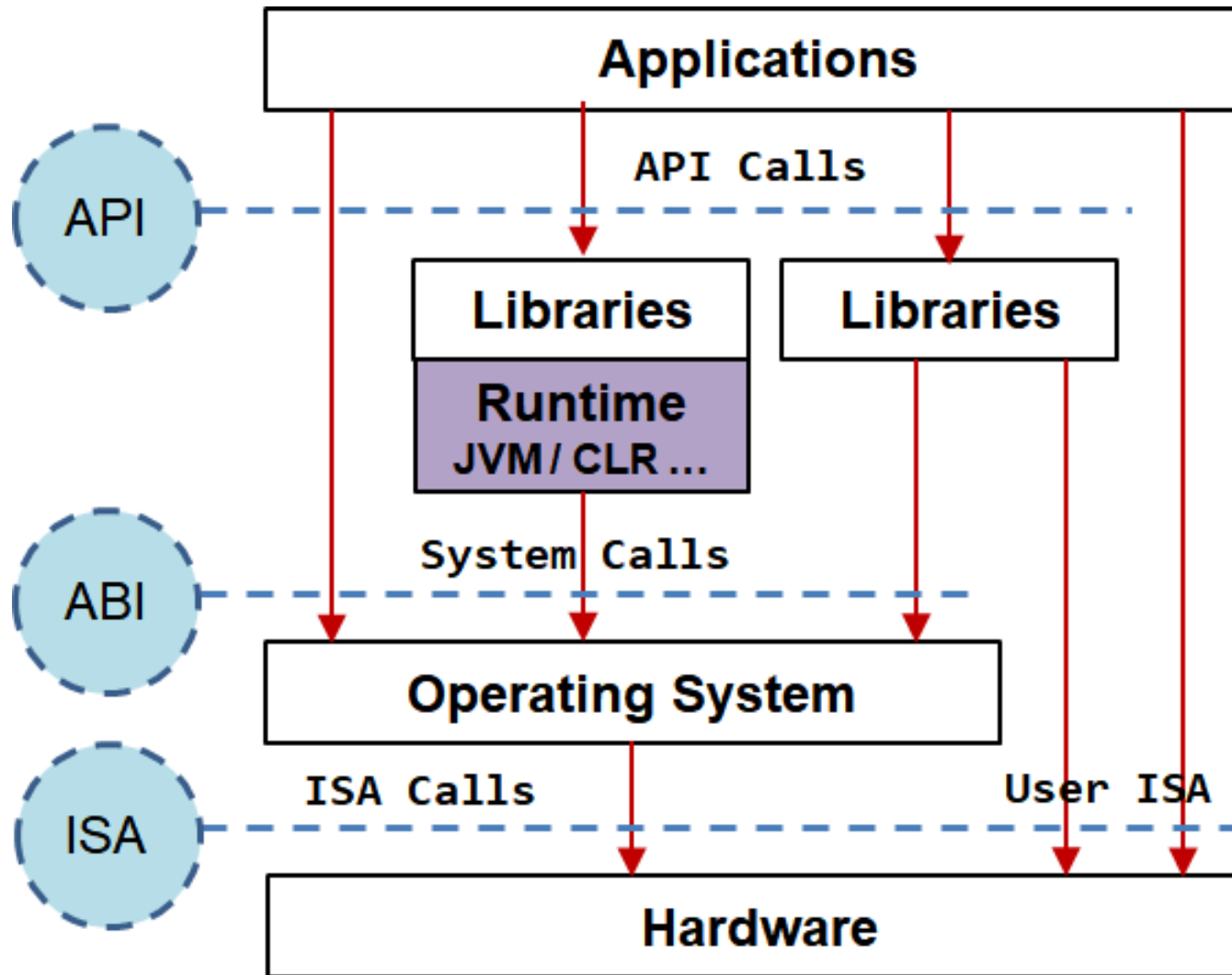
Porquê a virtualização?

- Rapidez de provisionamento e escalabilidade
 - Novas instâncias de VMs podem ser criadas a partir de configurações pré-definidas
 - Para melhor acomodar um aumento de carga (ex: número de pedidos) novas VMs podem ser colocadas em execução
- Migração e balanceamento de carga
 - Migração de VMs para consolidar e otimizar a utilização de hardware
 - É possível migrar VMs para outros recursos físicos com maior capacidade (ex: mais CPU, mais memória)
- Backups e recuperação de desastres
 - Facilita as tarefas de gestão das infraestruturas, nomeadamente backups e recuperação de desastres já que pode ser feito backup e recuperação da “imagem” do sistema num determinado momento

Interfaces de um sistema computacional

- A arquitetura de um sistema computacional determina as diferentes interfaces que existem para comunicação entre camadas
- Existem tipicamente 3 interfaces relevantes para aspetos de virtualização:
 - ISA: *Instruction Set Architecture*
 - Marca a separação entre hardware e software. Existem normalmente dois subconjuntos: uma ISA privilegiada e outra que pode ser invocada diretamente pelas aplicações
 - ABI: *Application Binary Interface*
 - Expõe os recursos com validação e controlo de acessos através de chamadas ao sistema operativo
 - API: *Application Programming Interface*
 - Interface de alto-nível que garante maior portabilidade para outros sistemas que suportam a mesma API

Interfaces de um sistema computacional

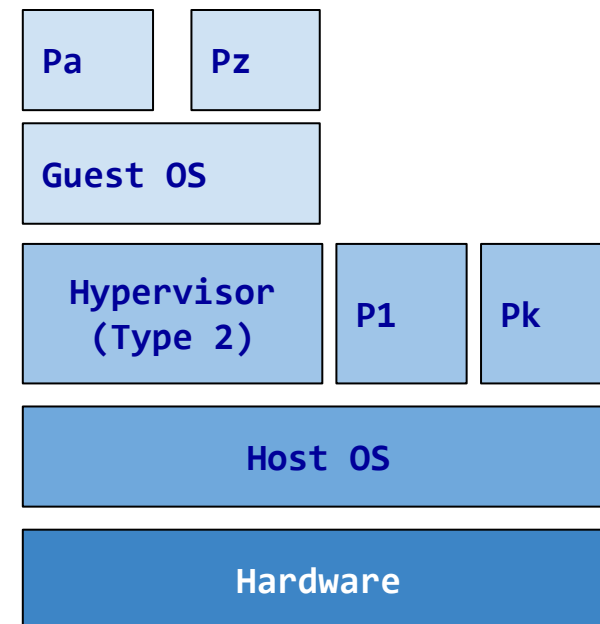
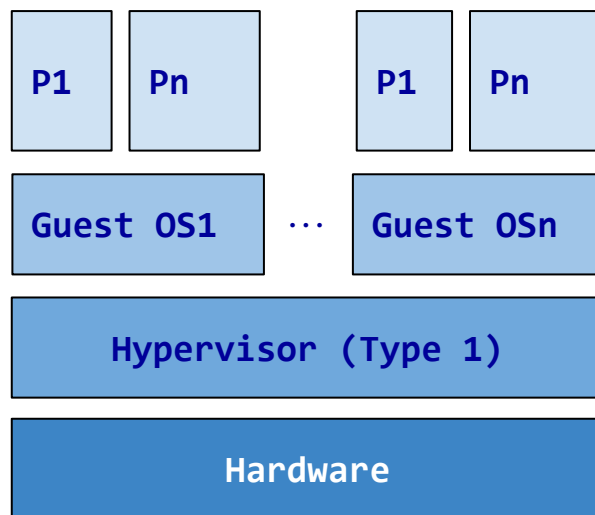


Máquinas virtuais de processo ou sistema

- O que é a “máquina virtual” depende da perspetiva do processo ou do sistema operativo
 - Na perspetiva do processo, a máquina é representada pela ABI, e na perspetiva da aplicação é representada pela API
 - Na perspetiva do SO, a máquina é representado pela ISA
- O software que suporta uma “máquina virtual” de processo é designado de *runtime*
- O software que suporta uma “máquina virtual” de sistema (VM) é referido como *virtual machine monitor* (VMM) ou *hypervisor*.
- O sistema operativo de uma VM é o **guest** (convidado)
- O software que suporta a VM é o **host** (hospedeiro)

Hypervisor e execução privilegiada

- O *hypervisor* é classificado como do Tipo 1 ou do Tipo 2, se respetivamente não depende, ou depende, da existência de um sistema operativo
- O tipo 1 (ou *bare metal*) interage diretamente com o *hardware* e não necessita de um sistema operativo, introduzindo menos *overhead*. Exemplos: Citrix/Xen, VMware ESXi; Microsoft Hyper-V
- O tipo 2 corre sobre um sistema operativo, tirando partido da transparência que o sistema tem a diferentes *hardwares*. Exemplos: Microsoft Virtual PC, Oracle Virtual Box, VMware Workstation; KVM



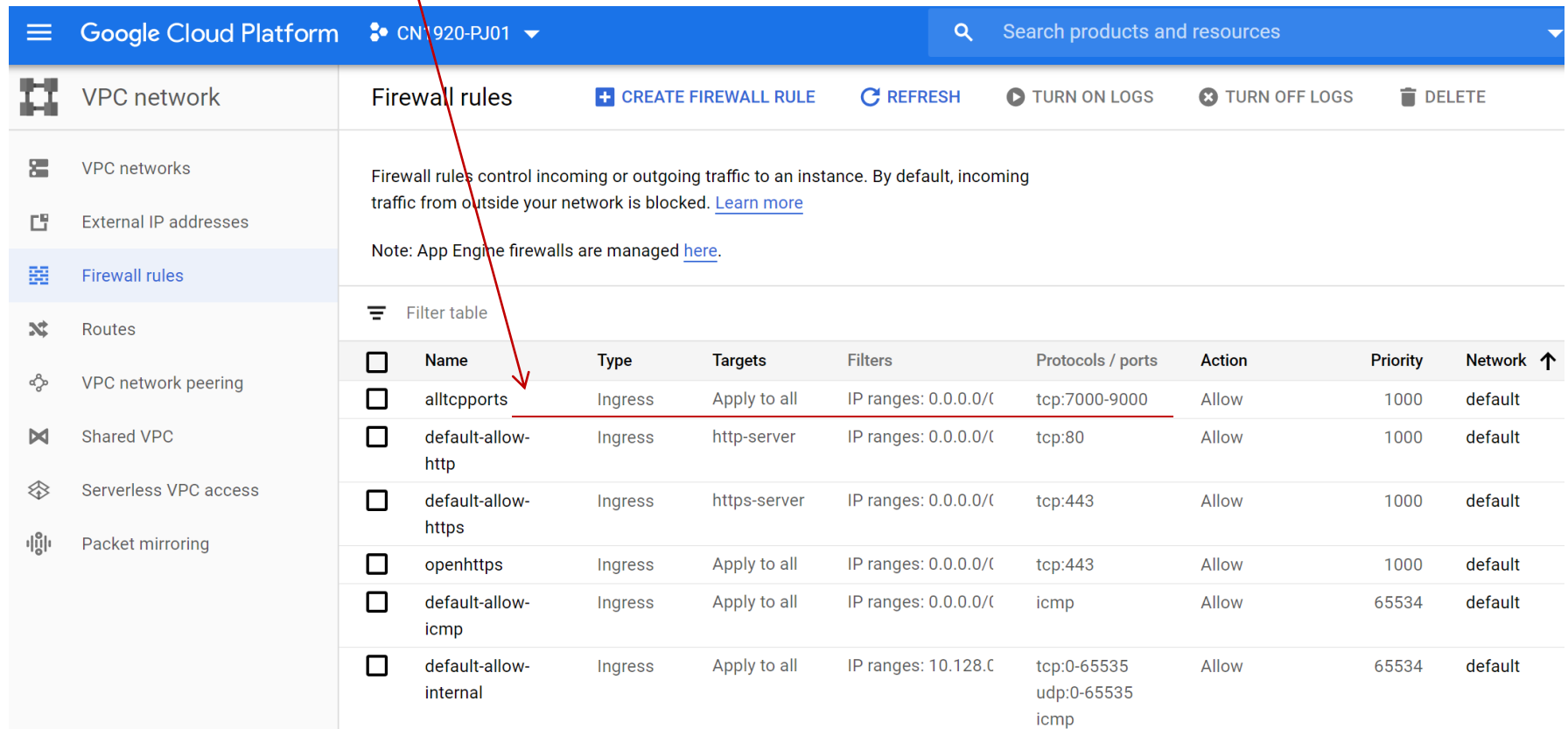
Serviço *Google Compute Engine*

Gestão de VMs na GCP

- **Firewalls**
- **Instâncias**
 - Tipo de instância (CPU, Memória)
 - Rede
 - Segurança
 - Discos referem imagens
- **Imagens**
 - Pré-definidas no GCP
 - Construídas a partir do disco de VMs existentes
- **Template**
 - Configuração pré-definida de uma instância
- **Grupos de instâncias**
 - Agrupamento de instâncias a partir de *templates*
 - Aumento ou redução do número de instâncias (elasticidade)

Regras de *firewall*

- Cada projeto define regras de *firewall* que podem ser aplicadas a diferentes VMs e diferentes intervalos de endereços
- Existe um conjunto base de regras, que pode ser modificado, por exemplo, criar novas regras



Google Cloud Platform CN 920-PJ01

Search products and resources

VPC network

Firewall rules [+ CREATE FIREWALL RULE](#) [REFRESH](#) [TURN ON LOGS](#) [TURN OFF LOGS](#) [DELETE](#)

VPC networks

External IP addresses

Firewall rules

Routes

VPC network peering

Shared VPC

Serverless VPC access

Packet mirroring

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)

Note: App Engine firewalls are managed [here](#).

Filter table

<input type="checkbox"/>	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network	↑
<input type="checkbox"/>	alltcpports	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:7000-9000	Allow	1000	default	
<input type="checkbox"/>	default-allow-http	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:80	Allow	1000	default	
<input type="checkbox"/>	default-allow-https	Ingress	https-server	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000	default	
<input type="checkbox"/>	openhttps	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000	default	
<input type="checkbox"/>	default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default	
<input type="checkbox"/>	default-allow-internal	Ingress	Apply to all	IP ranges: 10.128.0.0/16	tcp:0-65535 udp:0-65535 icmp	Allow	65534	default	

Definição de nova regra de *firewall*

← Create a firewall rule

Targets

All instances in the network

Source filter

IP ranges

Source IP ranges *

0.0.0.0/0 for example, 0.0.0.0/0, 192.168.2.0/24

Second source filter

None

Protocols and ports ?

☐ Allow all

☒ Specified protocols and ports

☒ tcp :

7000-9000

☐ udp :

all

☐ Other protocols

protocols, comma separated, e.g. ah, sctp

▼ DISABLE RULE

CREATE

CANCEL

Criar uma VM base

❖ Criar uma instância de VM:

- ✓ Nome: **vm-centos8-with-stress**
- ✓ Serie: **E2** Tipo: **e2-micro (2 vCPU, 1 GB memory)**
- ✓ Em Boot Disk escolha o sistema operativo: **CentOS Stream 8**
- ✓ Colocar as chaves SSH para acesso com o cliente SSH (ex: Bitvise)

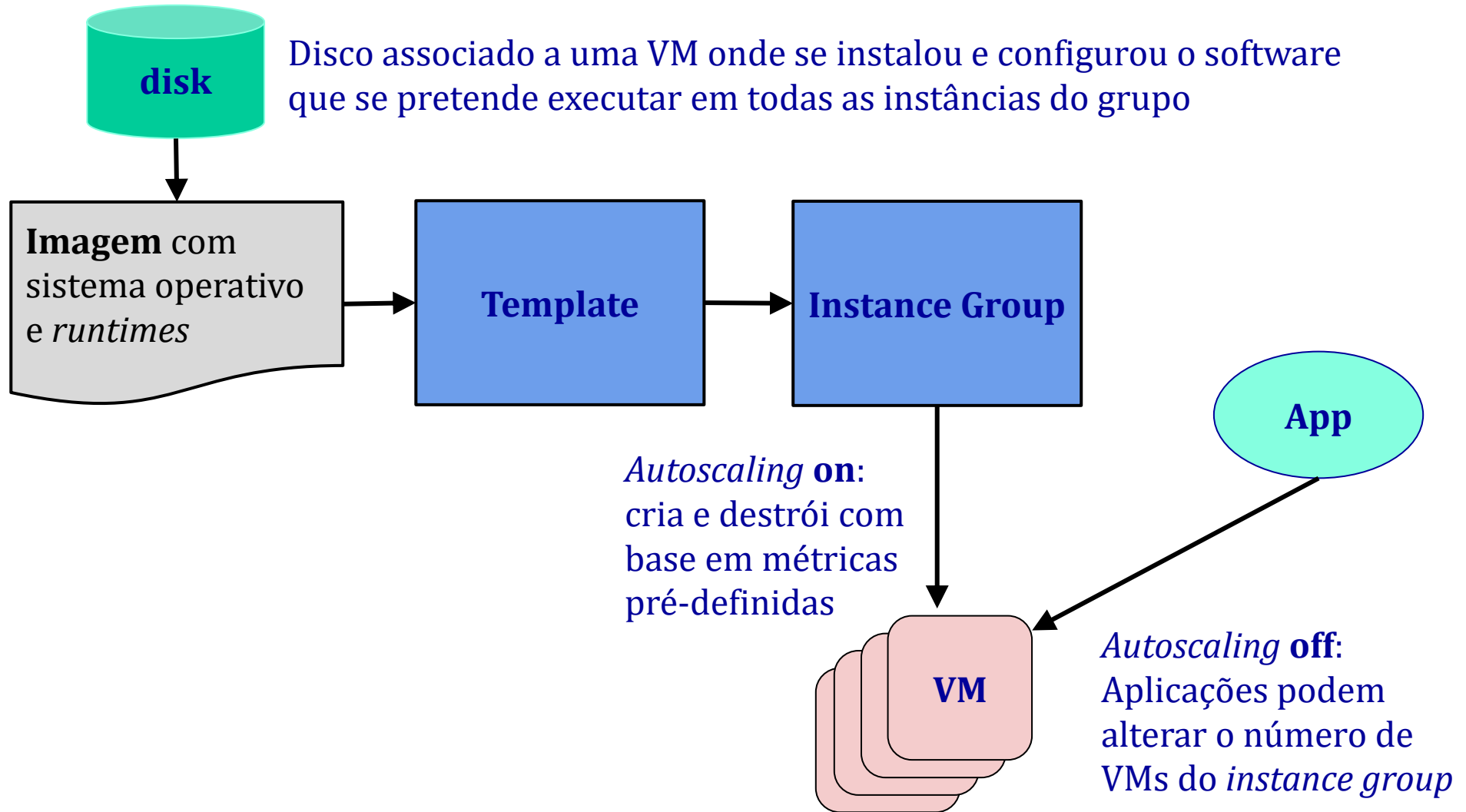
❖ Após arranque e utilizando o cliente SSH:

- ✓ Instalar Java 11: **sudo yum install java-11-openjdk-devel**
- ✓ Instalar utilitário stress:
 - ✓ **sudo yum install -y epel-release**
 - ✓ **sudo yum install -y stress**
- ✓ Instalar software base, ex: servidores, BD, aplicações; etc.

❖ Verifique nos detalhes da VM o nome do disco associado à VM

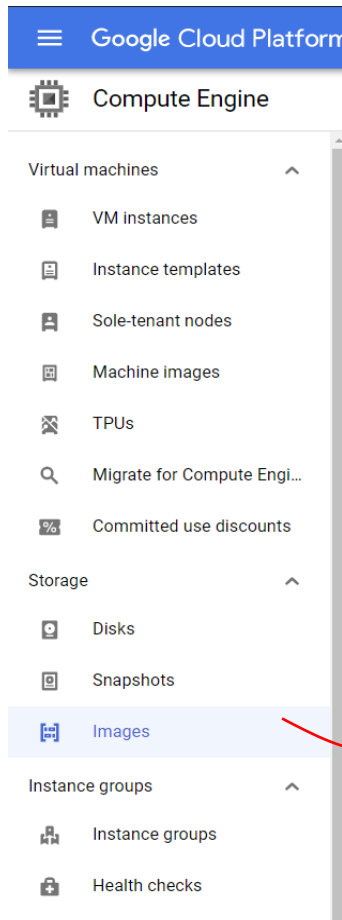
Boot disk				
Name	Image	Size (GB)	Device name	Type
vm-centos8-with-stress	centos-8-v20210512	20	vm-centos8-with-stress	Balanced persistent disk

Controlo da elasticidade de VMs



Criar nova imagem a partir do disco de uma VM

A instância de VM deve estar *Stopped*



Create an image

Name [?]
Name is permanent
image-centos8-with-stress

Source [?]
Disk

Source disk [?]
vm-centos8-with-stress

Location [?]
☐ Multi-regional
☒ Regional
europe-west2 (London) (default)

Family (Optional) [?]

Description (Optional)

Labels [?] (Optional)
[+ Add label](#)

Encryption
Data is encrypted automatically. Select an encryption key management solution.
☒ Google-managed key
No configuration required
☐ Customer-managed key
Manage via Google Cloud Key Management Service
☐ Customer-supplied key
Manage outside of Google Cloud

You will be billed for this image. [Compute Engine pricing](#)

Create Cancel

Lista de imagens

- Na lista de imagens do projeto GC, passa a existir mais uma imagem para além daquelas que já existiam e podíamos usar na criação de VMs

The screenshot shows the Google Cloud Platform console interface. On the left, the 'Compute Engine' section is expanded, showing 'Virtual machines' and 'Machine images'. The 'Images' page is active, displaying a table of images for the project 'PJBASE-CN2021'. The table has columns for Status, Name, Location, Archive size, Disk size, Created by, and Failures. The image 'image-centos8-with-stress' is highlighted with a red box, indicating it is the new image added to the list.

Status	Name	Location	Archive size	Disk size	Created by	Failures
✓	image-centos8-jdk11-nmap	us-central1	1.05 GB	20 GB	pjbase-cn2021	
✓	image-centos8-with-stress	europa-west2	1.2 GB	20 GB	pjbase-cn2021	
✓	c0-deeplearning-common-cu110-v20210512-debian-10	asia, eu, us	—	50 GB	Debian	co gp
✓	c1-deeplearning-tf-1-15-cu110-v20210512-debian-10	asia, eu, us	—	50 GB	Debian	tf- de
✓	c1-deeplearning-tf-1-15-tpu-v20210512-debian-10	asia, eu, us	—	50 GB	Debian	tf- de
✓	c1-deeplearning-tf-2-1-cu110-v20210512-debian-10	asia, eu, us	—	50 GB	Debian	tf2 de
✓	c1-deeplearning-tf-2-1-tpu-v20210512-debian-10	asia, eu, us	—	50 GB	Debian	tf- de
✓	c1-deeplearning-tf-2-3-cu110-v20210512-debian-10	asia, eu, us	—	50 GB	Debian	tf2 de
✓	c1-deeplearning-tf-2-3-tpu-v20210512-debian-10	asia, eu, us	—	50 GB	Debian	tf- de

Templates

- Um *template* é criado como uma instância de VM, seleccionando como *Boot disk* a imagem anteriormente criada. Assim, tal como se faz para uma VM, deve configurar-se: número de CPUs, memória, rede, segurança com chaves SSH, etc.

Boot disk ?



New 20 GB balanced persistent disk

Image

image-centos8-with-stress

Change

Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#)

PUBLIC IMAGES

CUSTOM IMAGES

SELECT A PROJECT

Show images from: cn2122-jsla-geral

☐ Show deprecated images

Image *

image-slides-centos8-stress

Boot disk type *

Balanced persistent disk

Size (GB) *

20

Google Cloud Platform

PJBASE-CN2021

Search products and resources

Compute Engine

Instance templates

CREATE INSTANCE TEMPLATE

REFRESH

CREATE INSTANCE GROUP

COPY

DELETE

Virtual machines

VM instances

Instance templates

Sole-tenant nodes

Filter

Filter instance templates

	Name	Machine type	Image	Disk type	Placement policy	In use by	Creation time	Actions
	template-centos7-with-stress	e2-micro	centos-7-v20210512	Balanced persistent disk	No policy	instance-group-1	May 20, 2021, 6:10:33 PM UTC+01:00	
	template-centos8-with-stress	e2-micro	image-centos8-with-stress	Balanced persistent disk	No policy		May 21, 2021, 11:01:11 PM UTC+01:00	

Instance Group

- O *Compute Engine* oferece dois tipos de grupos de instâncias de VM, *managed* e *unmanaged*:
 - Grupos de instâncias *managed*
 - Criadas a partir de um template, partilhando por isso um conjunto comum de configurações
 - Grupos de instâncias *unmanaged*
 - Grupos de VMs que não partilham o mesmo *template*
- Os grupos de instâncias *managed* têm elasticidade para aumentar/diminuir o número de instâncias entre um número mínimo e um número máximo
- Grupos *managed* podem ser usados em cenários *stateless* ou *stateful*:
 - ***Stateless***
 - **Processamento de pedidos obtendo o pedido a partir de filas de mensagens (por exemplo, Pub/Sub)**
 - *Stateful*
 - Aplicações *stateful* com processamento de tarefas de longa duração que precisem de persistir estado intermédio

Managed Instance Group (stateless)

← Create Instance Group



New managed instance group (stateless)

Automatically manage groups of VMs that do stateless serving and batch processing.

Set up automatic management for a group of stateless VMs, including updates, regional deployments, load balancing, autoscaling, and autohealing. [Learn more](#)

Name *
instance-group-slides-centos8-stress ?

Name is permanent

Description

Instance template *
template-slides-vm-centos8-with-stress ▾ ?

Number of instances

Based on autoscaling configuration

Location

For higher availability, select multiple zones in a region instead of a single zone. [Learn more](#)

☒ Single zone

☐ Multiple zones

Region *
europe-west1 (Belgium) ▾ ?

Zone *
europe-west1-b ▾ ?

Autoscaling

Use autoscaling to automatically add and remove instances to the group for periods of high and low load. [Learn more](#)

Autoscaling mode
On: add and remove instances to the group ▾

Minimum number of instances *
1 ?

Maximum number of instances *
5 ?

Nome do grupo a criar

Nome do *template*
(previamente criado) com
configuração base de novas
instâncias

Zona onde serão criadas as
instâncias

Autoscaling On: O aumento e
diminuição de instâncias é feito
automaticamente baseado em métricas

Número mínimo (inicial) e
máximo de instâncias

Autoscaling On

Autoscaling mode

On: add and remove instances to the group

Scale out: only add instances to the group

Off: do not autoscale

Autoscaling Off: O aumento e diminuição de instâncias é feito manualmente na consola ou através de uma aplicação

- Caso o **autoscaling On** esteja ativo, podem ser definidas métricas com base em:
 - Utilização de CPU
 - HTTP load balancing
 - Métricas de monitorização

Autoscaling

Use autoscaling to automatically add and remove instances to the group for periods of high and low load. [Learn more](#)

Autoscaling mode

On: add and remove instances to the group

Minimum number of instances *

1



Maximum number of instances *

5



Autoscaling metrics

Use metrics to help determine when to scale the group. [Learn more](#)

Edit metric

Metric type *

CPU utilization

Target CPU utilization *

60

%



Predictive autoscaling

Use predictive autoscaling to predict future capacity needs based on historical load.



To ensure accuracy, predictive autoscaling will collect data about this managed instance group for at least 3 days before starting to forecast its size.

☒ Off

The autoscaler scales the group to meet current demand based on real-time metrics.

☐ Optimize for availability

Predictive autoscaling improves availability by monitoring daily and weekly load patterns and scaling out ahead of anticipated demand.

Um exemplo

- *Managed Instance Group* com **Autoscaling On**
- Intervalo de número de instâncias de VMs: 1 (mínimo) a 4 (máximo)
- CPU *capacity*:
 - Inicialmente de 50%
 - Ao fim de ~40min. ajustado para 60%
- Cada instância de VM tem instalada a aplicação de *stress*
 - <https://linux.die.net/man/1/stress>
 - O comando `stress -cpu N` lança N processos, cada um a executar uma tarefa que consome CPU de forma intensiva.
- **Observações:**
 - Muito rápido a aumentar o número de instâncias, lento a diminuir

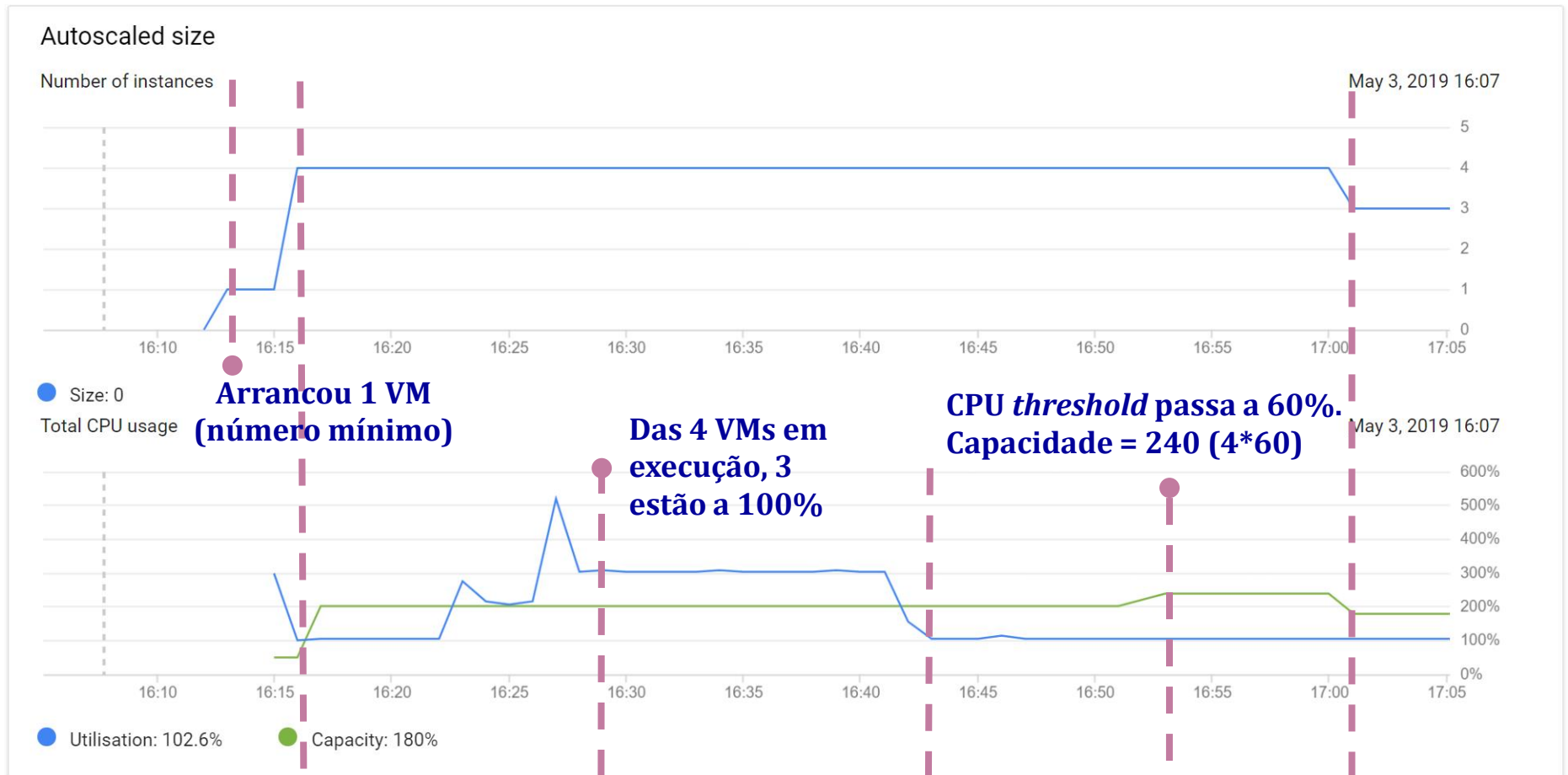
Managed Group com autoscalling On



Para aumentar a utilização de CPU usou-se o *stress*

capacidade = nº VMs x *CPU threshold*
No caso capacidade = 200 (4 * 50)

Managed Group com autoscalling On



A intensa utilização inicial (utilização acima da capacidade) desencadeou a criação de novas instâncias (+3)

Apenas 1 VM a 100%

Grupo é reduzido para 3 instâncias
Capacidade = 180 (3*60)

Scripts de *startup* e *shutdown*

Management Security Disks Networking Sole Tenancy

Description (Optional)

Labels ? (Optional)

+ Add label

Reservations

Automatically use created reservation

Automation

Startup script (Optional)

You can choose to specify a startup script that will run when your instance boots up or restarts. Startup scripts can be used to install software and updates, and to ensure that services are running within the virtual machine. [Learn more](#)

```
#!/bin/bash
date > /tmp/startuptime.txt
```

Metadata (Optional)

You can set custom metadata for an instance or project outside of the server-defined metadata. This is useful for passing in arbitrary values to your project or instance that can be queried by your code on the instance. [Learn more](#)

shutdown-script

```
#!/bin/bash
date > /tmp/shutdowntime.txt
```

+ Add item

- É frequente a necessidade de fazer ações de *startup* e *shutdown* de instâncias
- Tal pode ser feito:
 - através de scripts configurados na imagem, usando comandos específicos da versão do sistema operativo
 - através de scripts associados ao *template*, que são executados pelo GCP no arranque e na desativação das instâncias

<https://cloud.google.com/compute/docs/startupscript>

<https://cloud.google.com/compute/docs/shutdownscript>

API Java para o Compute Engine

- Javadoc:

<https://googleapis.dev/java/google-cloud-compute/latest/index.html>

- Repositório de Código:

<https://github.com/googleapis/java-compute>

- Autenticação/Autorização com variável de ambiente

- GOOGLE_APPLICATION_CREDENTIALS
- Role “Compute Admin”

```
<dependency>  
  <groupId>com.google.cloud</groupId>  
  <artifactId>google-cloud-compute</artifactId>  
  <version>1.9.1</version>  
</dependency>
```

Listagem de VM numa zona

```
static void listVMInstances(String project, String zone) throws IOException {  
    System.out.println("=== Listing VM instances at zone: "+zone);  
    try (InstancesClient client = InstancesClient.create()) {  
        for (Instance instance : client.list(project, zone).iterateAll()) {  
            System.out.println("Name: " + instance.getName() + " VMId:" + instance.getId());  
            System.out.println("  Network interfaces:"+instance.getNetworkInterfacesCount());  
            String ip = instance.getNetworkInterfaces(0).getAccessConfigs(0).getNatIP();  
            System.out.println("  IP: " + ip);  
            System.out.println("  Status: " + instance.getStatus() +  
                               " : Last Start time: " + instance.getLastStartTimestamp());  
        }  
    }  
}
```

```
Name: lab4-forum-server VMId:2040227344145152434  
Numero de network interfaces: 1  
IP: 104.199.100.141  
Status: TERMINATED : Last Start time: 2022-05-12T15:20:51.892-07:00  
Name: slides-vm-centos8-with-stress VMId:4469506343351848363  
Numero de network interfaces: 1  
IP: 34.140.241.95  
Status: RUNNING : Last Start time: 2022-05-14T15:31:05.192-07:00
```

Start/Stop VM

```
static void startStoppedVM(String projectID, String zone, String instanceName) {  
    try (InstancesClient client = InstancesClient.create()) {  
        StartInstanceRequest request =  
            StartInstanceRequest.newBuilder()  
                .setProject(projectID)  
                .setZone(zone)  
                .setInstance(instanceName)  
                .build();  
        OperationFuture<Operation, Operation> fut = client.startAsync(request);  
        while (!fut.isDone()) {  
            System.out.println("waiting to start!");  
            Thread.sleep(4 * 1000);  
        }  
        System.out.println("'" + fut.get().getStatus().toString());  
    }  
}
```

```
// to stop  
.stopAsync(request)
```

Listagem de *instance groups*

```
void listManagedInstanceGroups(String project, String zone) throws IOException {  
    try (InstanceGroupManagersClient client = InstanceGroupManagersClient.create()) {  
        for (InstanceGroupManager manager : client.list(project, zone).iterateAll()) {  
            System.out.println("Name: " + manager.getName());  
            System.out.println("Template: " + manager.getInstanceTemplate());  
        }  
    }  
}
```

Name: instance-group-1

Template: <https://www.googleapis.com/compute/v1/projects/pjbase-cn2021/global/instanceTemplates/template-centos8-with-stress>

Listagem de VM de um *instance group*

```
static void listManagedInstanceGroupVMs(String projectId, String zone, String grpName) {
    InstanceGroupManagersClient managersClient = InstanceGroupManagersClient.create();
    ListManagedInstancesInstanceGroupManagersRequest request =
        ListManagedInstancesInstanceGroupManagersRequest.newBuilder()
            .setInstanceGroupManager(grpName)
            .setProject(projectId)
            .setReturnPartialSuccess(true)
            .setZone(zone)
            .build();
    System.out.println("Instances of instance group: " + grpName);
    for (ManagedInstance instance :
        managersClient.listManagedInstances(request).iterateAll()) {
        System.out.println(instance.getInstance()+"STATUS =" + instance.getInstanceStatus());
    }
}
```


Redimensionar instâncias de um *instance group*

```
static void resizeManagedInstanceGroup(String project, String zone, String instanceGroupName,
int newSize) {
    System.out.println("===== Resizing instance group");
    InstanceGroupManagersClient managersClient = InstanceGroupManagersClient.create();
    OperationFuture<Operation, Operation> result = managersClient.resizeAsync(
        project,
        zone,
        instanceGroupName,
        newSize
    );
    Operation oper=result.get();
    System.out.println("Resizing with status " + oper.getStatus().toString());
}
```