TÉCNICO LISBOA

Enterprise Integration

Containers & Cloud

Prof. Sérgio Guerreiro
Sergio.guerreiro@tecnico.ulisboa.pt

Department of Computer Science and Engineering
Instituto Superior Técnico / Universidade de Lisboa
INESC-ID
URL: http://www.inesc-id.pt
Rua Alves Redol, 9
1000-029 Lisboa
Portugal

TÉCNICO LISBOA

Cloud services properties

- **Broadband access** - Consume the services from anywhere
- **On-demand self-service** - Consume the services when you want
- **Resource pooling and virtualization** - Pool the infrastructure, virtual platforms and applications
- **Rapid elasticity** - Pooled resources with horizontal scalability
- **Measured service** - Pay only for what you consume when you consume

Cloud services

SaaS	Software As a Service   
PaaS	Platform As a Service  
IaaS	Infrastructure As a Service  
IaC	Infrastructure as Code 

IaaS vs. PaaS vs. SaaS

You have more complete control over the configuration of your cloud resources in IaaS than in PaaS and SaaS. PaaS and SaaS virtualize more infrastructure functions and you have fewer components to manage as compared to IaaS.

Consider the following table. If you manage your own IT infrastructure, you must invest in and maintain all the items mentioned in the table. On the other hand, if you switch to a particular cloud computing service, this is how it works:

- Yellow cells indicate what you manage
- Green cells indicate what the cloud service provider manages

	Own IT	IaaS	PaaS	SaaS
Application	Y	Y	Y	Y
Data	Y	Y	Y	G
Runtime or software that runs the application	Y	Y	G	G
Middleware or software that monitors the application	Y	Y	G	G
Operating systems on which the application runs	Y	G	G	G
Virtualization technology	Y	G	G	G
Server machines	Y	G	G	G
Storage devices	Y	G	G	G
Network appliances	Y	G	G	G

https://aws.amazon.com/what-is/iaas/

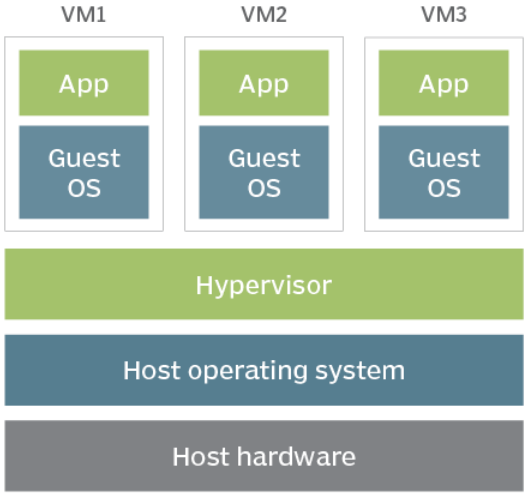
Clouds may be hosted and employed in different styles depending on the use case, respectively the business model of the provider: **Private** cloud; **Community** cloud; **Public** cloud; **Hybrid** cloud; **Special purpose** clouds.

Physical Servers

- Slow-iteration and slow-deployment
- Single tenancy
- Unfriendly for friendly for multi programming languages
- Deploy in weeks
- Typically, alive for years

Virtual Machines (VMs)

- VMs work by operating on top of a hypervisor, which is stacked on top of a host machine
- A VM monitor (VMM) or **hypervisor** intermediates between the host and guest VM. By isolating individual guest VMs from each other, the VMM enables a host to support multiple guests running different OSes.
- Each VM carries their own virtualized hardware stack that comprises network adapters, storage, applications, binaries, libraries and its own CPU
- Advantages:
  - VMs allow to consolidate applications onto a single server.
  - Faster-iteration and deployment
  - Multi-tenancy
  - Somewhat friendly for multi programming languages
  - Deploy in minutes
  - Typically, alive for weeks
- Disadvantages: Having multiple VMs with their owned OS adds substantial **overheads** in terms of RAM, CPU, I/O and storage



Containers

- A container is a **self-contained** execution environment that shares the kernel of the host system and which is (optionally) isolated from other containers in the system
- One of the major advantages of containers is resource **efficiency**, because you don’t need a whole operating system instance for each isolated workload

- When a process is running inside a container, there is only a **little bit of code** that sits inside the kernel managing the container.
- Contrast this with a virtual machine where there would be a second layer running. In a VM, **calls by the process to the hardware or hypervisor** would require bouncing in and out of privileged mode on the processor twice, thereby noticeably slowing down many calls.

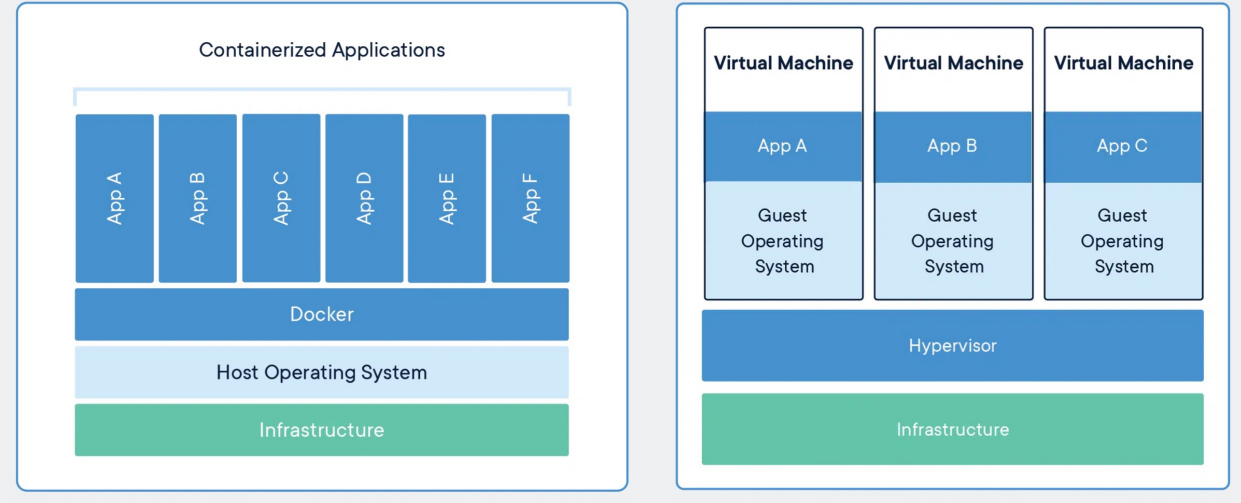
Kane, S. P., & Matthias, K. (2018). Docker: up & running: shipping reliable containers in production. O'Reilly Media.

Docker containerization

Containers are an **abstraction at the app layer** that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating systems.

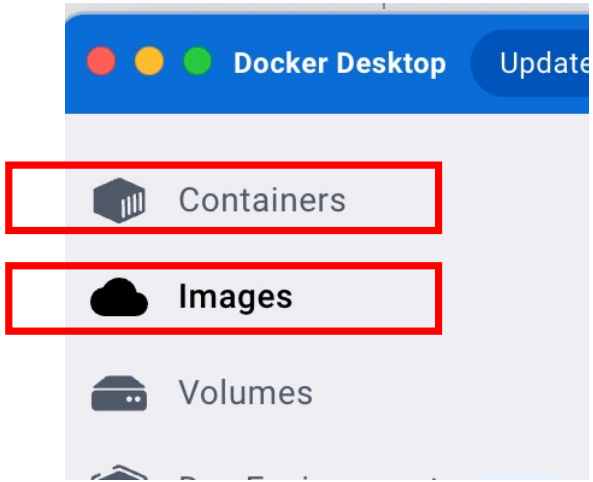
Advantage: more lightweight than VM's

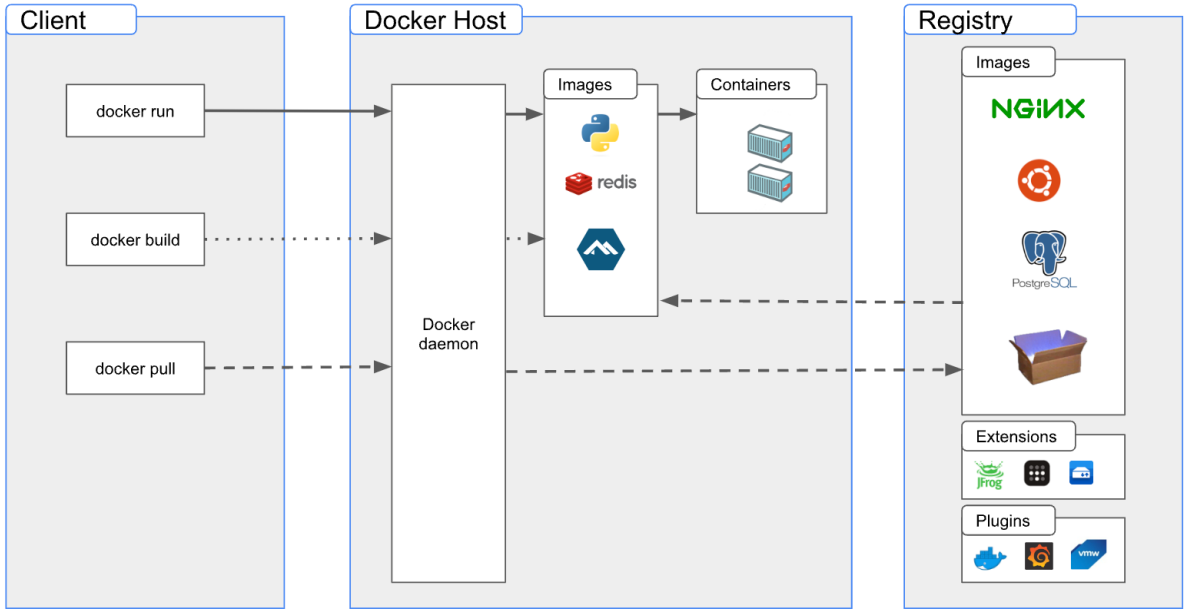
Containers sit on top of a Docker engine.



Kane, S. P., & Matthias, K. (2018). Docker: up & running: shipping reliable containers in production. O'Reilly Media.

Docker core concepts



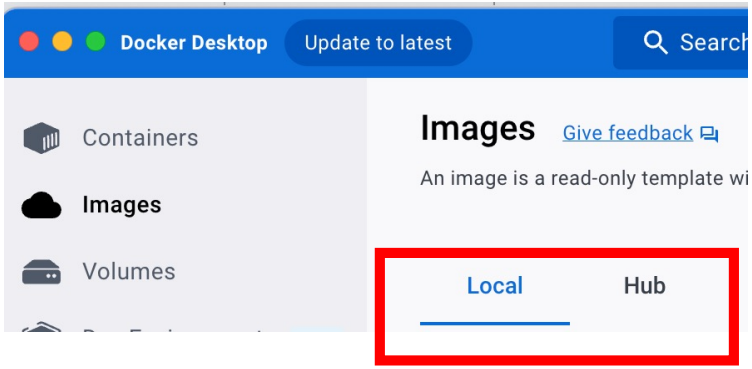


Kane, S. P., & Matthias, K. (2018). Docker: up & running: shipping reliable containers in production. O'Reilly Media.

Docker **image** *definition*

- An *image* is a **read-only template** with instructions for creating a Docker container. Often, an image is *based on* another image, with some additional customization.
  - For example, you may build an image which is based on the ubuntu image, but installs the Apache web server and your application, as well as the configuration details needed to make your application run.

- You might create your **own images** or you might only use those created **by others** and published in a registry. To build your own image, you create a *Dockerfile* with a simple syntax for defining the steps needed to create the image and run it. Each instruction in a Dockerfile creates a layer in the image. When you change the Dockerfile and rebuild the image, only those layers which have changed are rebuilt. This is part of what makes images so lightweight, small, and fast, when compared to other virtualization technologies.



Kane, S. P., & Matthias, K. (2018). Docker: up & running: shipping reliable containers in production. O'Reilly Media.
https://docs.docker.com/get-started/overview/

Docker **container** *definition*

- A container is a runnable instance of an image. You can **create**, **start**, **stop**, **move**, or **delete** a container using the Docker API or CLI. You can connect a container to **one** or **more networks**, **attach storage** to it, or even **create a new image** based on its current state.
- By default, a container is relatively well isolated from other containers and its host machine. You can control **how isolated** a container's network, storage, or other underlying subsystems are from other containers or from the host machine.
- A container is defined by its image as well as any configuration options you provide to it when you create or start it.
- When a container is removed, any changes to its state that are not stored in persistent storage disappear.

Example docker run command

- The following command runs an ubuntu container, attaches interactively to your local command-line session, and runs /bin/bash.

```
$ docker run -i -t ubuntu /bin/bash
```

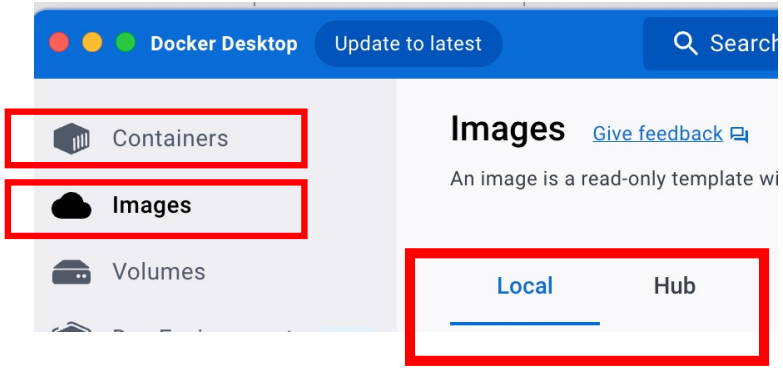
Kane, S. P., & Matthias, K. (2018). Docker: up & running: shipping reliable containers in production. O'Reilly Media.
https://docs.docker.com/get-started/overview/

Docker Hub

Is a service provided by Docker for finding and sharing container images with others. It's the world's largest repository of container images with an array of content sources including container community developers, open source projects and independent software vendors (ISV) building and distributing their code in containers.

Docker Hub provides the following major features:

- Repositories: Push and pull container images.
- Teams & Organizations: Manage access to private repositories of container images.
- Docker Official Images: Pull and use high-quality container images provided by Docker.
- Docker Verified Publisher Images: Pull and use high- quality container images provided by external vendors.
- Docker-Sponsored Open Source Images: Pull and use high- quality container images from non-commercial open source projects.
- Builds: Automatically build container images from GitHub and Bitbucket and push them to Docker Hub.
- Webhooks: Trigger actions after a successful push to a repository to integrate Docker Hub with other services.



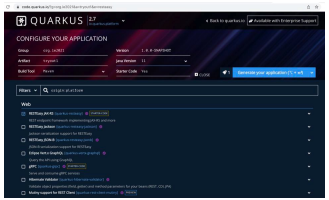
Kane, S. P., & Matthias, K. (2018). Docker: up & running: shipping reliable containers in production. O'Reilly Media.
https://docs.docker.com/docker-hub/

Docker instance inside EC2 instance by dockerhub!

TÉCNICO LISBOA

Test remotely:

Create the project: web or command line




```
mvn io.quarkus.platform:quarkus-maven-
plugin:2.7.5.Final:create \
-DprojectGroupId=org.acme \
-DprojectArtifactId=reactive-pg-client-quickstart \
-Dextensions="resteasy-jackson,reactive-pg-
client,resteasy-mutiny,quarkus-smallrye-openapi" \
-DnoCode
```

EDIT the code

eclipse

Add container-image-docker

docker

Build Docker-image & **Push**

./mvnw clean package

docker

dockerhub

amazon web services EC2

docker

Install docker

Pull image

Create docker container

Run it

Compile locally:

./mvnw quarkus:dev

Test locally:

Docker **compose** *definition*

- Is a tool for defining and running **multi-container Docker applications**.
- a YAML file is used to configure the application's services. Then, with a command, you create and start all the services from your configuration.

- Compose works in all environments: production, staging, development, testing, as well as CI workflows. It also has commands for managing the whole lifecycle of applications:
  - Start, stop, and rebuild services
  - View the status of running services
  - Stream the log output of running services
  - Run a one-off command on a service

Kane, S. P., & Matthias, K. (2018). Docker: up & running: shipping reliable containers in production. O'Reilly Media.
https://docs.docker.com/compose/

TÉCNICO LISBOA

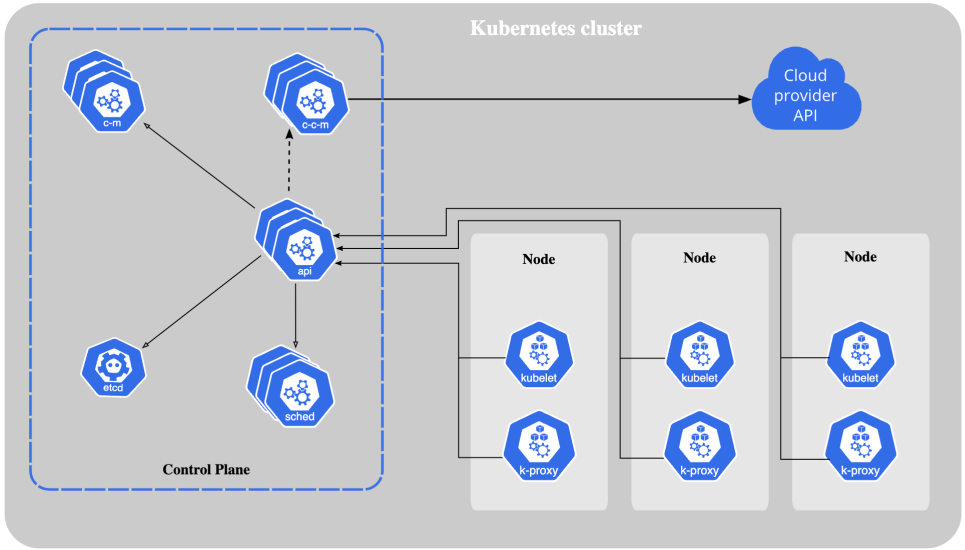
But, how to scale with more flexibility?

Kubernetes

- Is an open source orchestrator for deploying, scaling, and management of **containerized applications** *(different from terraform which is applicable to any cloud resource)*
- It allows to run Docker containers and workloads and helps tackling some of the operating complexities when moving to scale multiple containers, deployed across multiple servers
- The main benefits are
  - The service delivery velocity founded on
    - **Immutability**: rather than incremental updates and changes, an entirely new, complete image is built, where the update simply replaces the entire image with the newer imsge in a single operation
    - **Declarative configuration**: describing the state of the desired world, instead of specifying the series of instructions
    - **Online self-healing systems**: *e.g.*, if you assert a desired state of 3 replicas, kubernetes create exactly 3 replicas. If manually a fourth is created, kubernetes destroy it.
  - Scaling (software and teams)
    - Decoupling
    - Clusters
    - Microservices
    - Separation of concerns
  - Abstracting the infrastructure
  - Resource efficiency usage

Burns, B., Beda, J., Hightower, K., & Evenson, L. (2022). *Kubernetes: up and running*. " O'Reilly Media, Inc.".

Kubernetes Architecture



- A **node** is a machine where containers (workloads) are deployed. Every node in the cluster must run a container runtime such as Docker, as well as the below-mentioned components, for communication with the primary for network configuration of these containers
- A **Kubelet** is responsible for the running state of each node, ensuring that all containers on the node are healthy. It takes care of starting, stopping, and maintaining application containers organized into pods as directed by the control plane
- A **Kube-proxy** is an implementation of a network proxy and a load balancer, and it supports the service abstraction along with other networking operation It is responsible for routing traffic to the appropriate container based on IP and port number of the incoming request

https://kubernetes.io/docs/concepts/overview/components/

Kubernetes Pods

- A Pod represents a collection of application containers and volumes running in the same execution environment.
- A Pod is the smallest deployable artifact in a Kuberneted cluster
- All of the containers in a Pod always land on the same machine
- Applications running in the same Pod share the same IP address and port space (network namespace), have the same hostname, and can communicate using native interprocess ccommunication channels
- On the opposite, applications in different Pods are isolated from each other; they have different IP addresses, different hostnames,…

- Containers in different Pods running on the same node might as well be on different servers

- In general, "**Will these containers work correctly if they land on different machines?**"
  - **No** -> use a Pod to group the containers
  - **Yes** -> use multiple Pods

Burns, B., Beda, J., Hightower, K., & Evenson, L. (2022). *Kubernetes: up and running*. " O'Reilly Media, Inc.".

TÉCNICO LISBOA

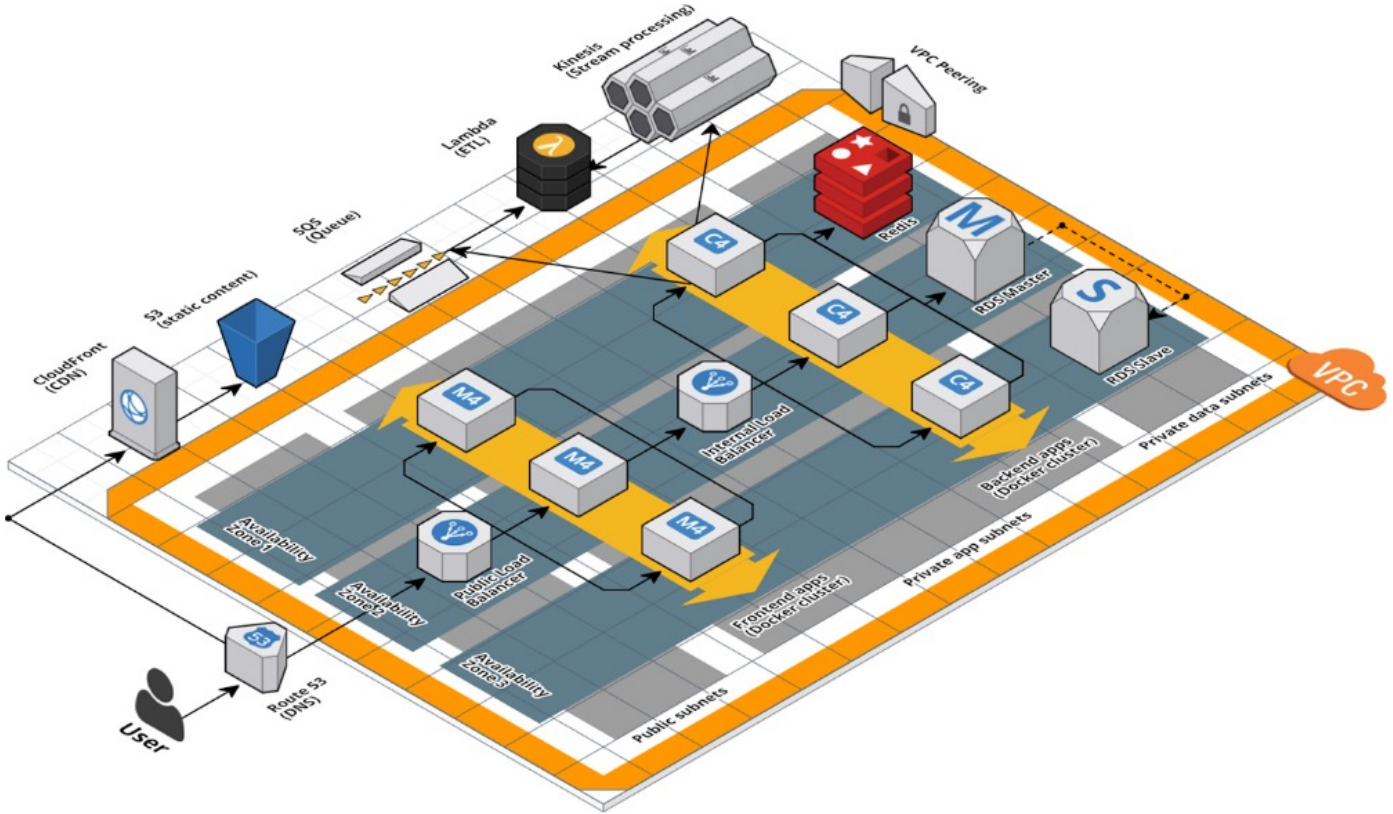
Other option?

Terraform – Infrastructure as Code (IaC)

- Terraform is a solution to create cloud environments using code instead of using the cloud providers User Interfaces.
- Therefore, the management of the cloud resources takes less effort and the creation, or update, process is faster

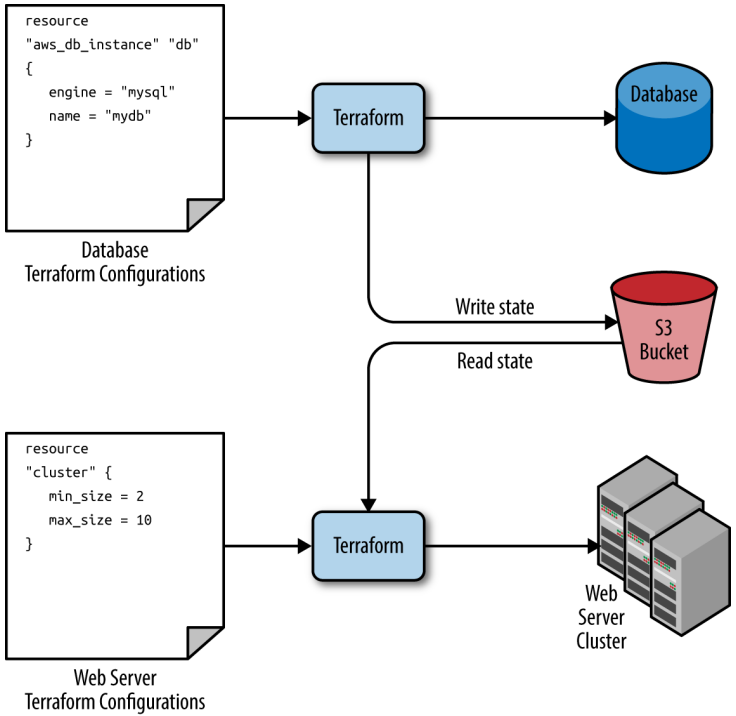
Brikman, Y. (2022). *Terraform: Up and Running*. " O'Reilly Media, Inc."

Terraform – Infrastructure as Code (IaC)



Brikman, Y. (2022). *Terraform: Up and Running*. " O'Reilly Media, Inc."

Terraform – State sharing



Brikman, Y. (2022). *Terraform: Up and Running*. " O'Reilly Media, Inc."

TÉCNICO LISBOA

But, I do not want to deploy the cluster, is it possible?

Serverless computing

- Serverless computing is a platform that hides server usage from developers and runs code on-demand automatically scaled and billed only for the time the code is running

- Cloud Native Computing Foundation (CNCF) defines serverless computing as *“the concept of building and running applications that do not require server management. It describes a finer grained deployment model where applications, bundled as one or more functions, are uploaded to a platform and then executed, scaled, and billed in response to the exact demand needed at the moment.”*

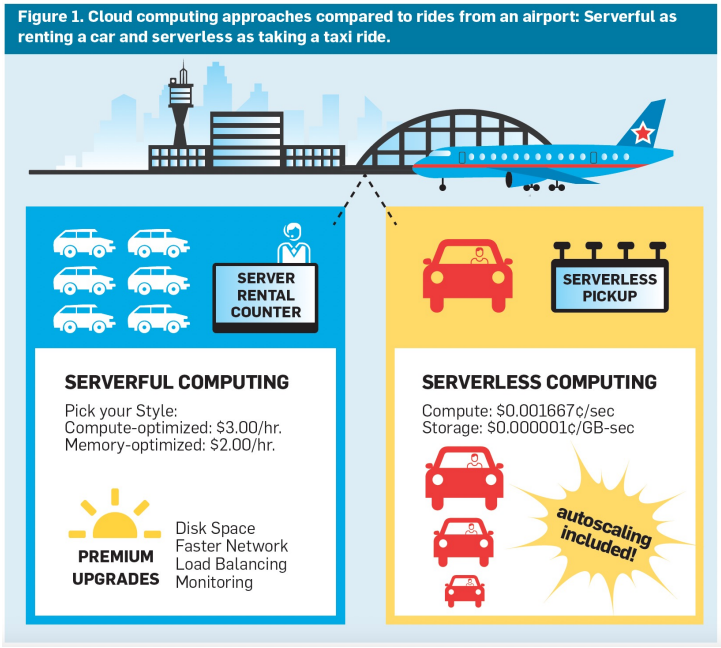
Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. (2019). The rise of serverless computing. Communications of the ACM, 62(12), 44-54.

Serverless essential qualities

1. Providing an **abstraction** that hides the servers and the complexity of programming and operating them

2. Offering a **pay-as-you-go cost model** instead of a reservation-based model, so there is no charge for idle resources

3. **Elasticity - Automatic**, **rapid**, and **unlimited scaling** resources up and down to match demand closely, from zero to practically infinite



Schleier-Smith, J., Sreekanti, V., Khandelwal, A., Carreira, J., Yadwadkar, N. J., Popa, R. A., ... & Patterson, D. A. (2021). What serverless computing is and should become: The next phase of cloud computing. *Communications of the ACM*, *64*(5), 76-84.

Resource usage: Serverless compared with Serverful

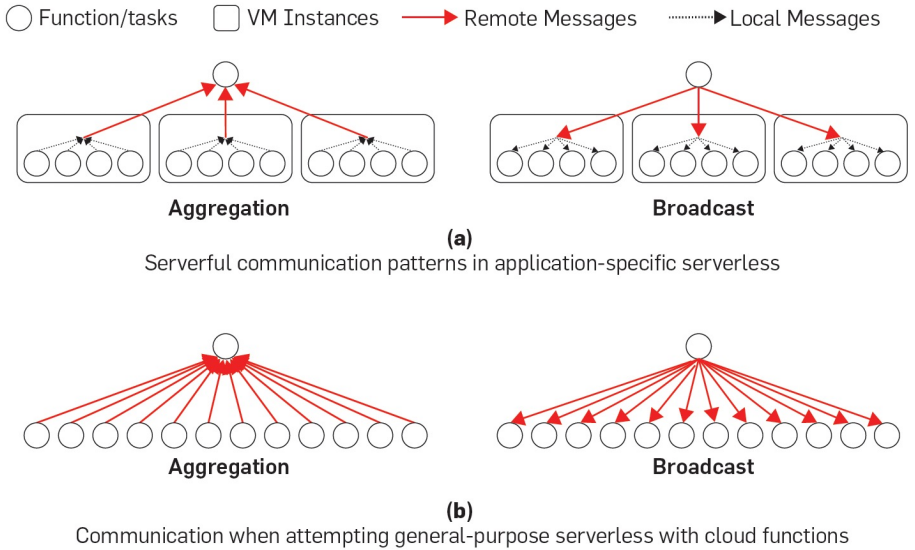
Figure 3. Serverless vs Serverful cloud computing: serverless users pay only for resources consumed, not for idle reserved capacity.

Schleier-Smith, J., Sreekanti, V., Khandelwal, A., Carreira, J., Yadwadkar, N. J., Popa, R. A., ... & Patterson, D. A. (2021). What serverless computing is and should become: The next phase of cloud computing. *Communications of the ACM*, *64*(5), 76-84.

Increased communication with Serverless

Figure 4. Increased communication for aggregation and broadcast patterns.

Application-specific serverless frameworks (for example, Cloud Dataflow) can be implemented with serverful communication patterns. In this case (a) the fewer arrows indicate less network communication than in (b) the general-purpose serverless option. By packing K tasks per VM instance, an application-specific serverless solution, like a serverful solution, is able to achieve a communication complexity of $O(N/K)$ for a job with N tasks, as opposed to $O(N)$ for the cloud function based alternative which can not influence task placement. Typical values for K range from 10 to 100, leading to an overall difference of one to two orders of magnitude.



Schleier-Smith, J., Sreekanti, V., Khandelwal, A., Carreira, J., Yadwadkar, N. J., Popa, R. A., ... & Patterson, D. A. (2021). What serverless computing is and should become: The next phase of cloud computing. *Communications of the ACM*, *64*(5), 76-84.

Function as a Service (FaaS)

- Function-as-a-Service is a serverless computing platform where the **unit of computation** is a function that is executed in response to triggers such as events or HTTP requests

Table 1. Comparison of different choices for cloud as a service.

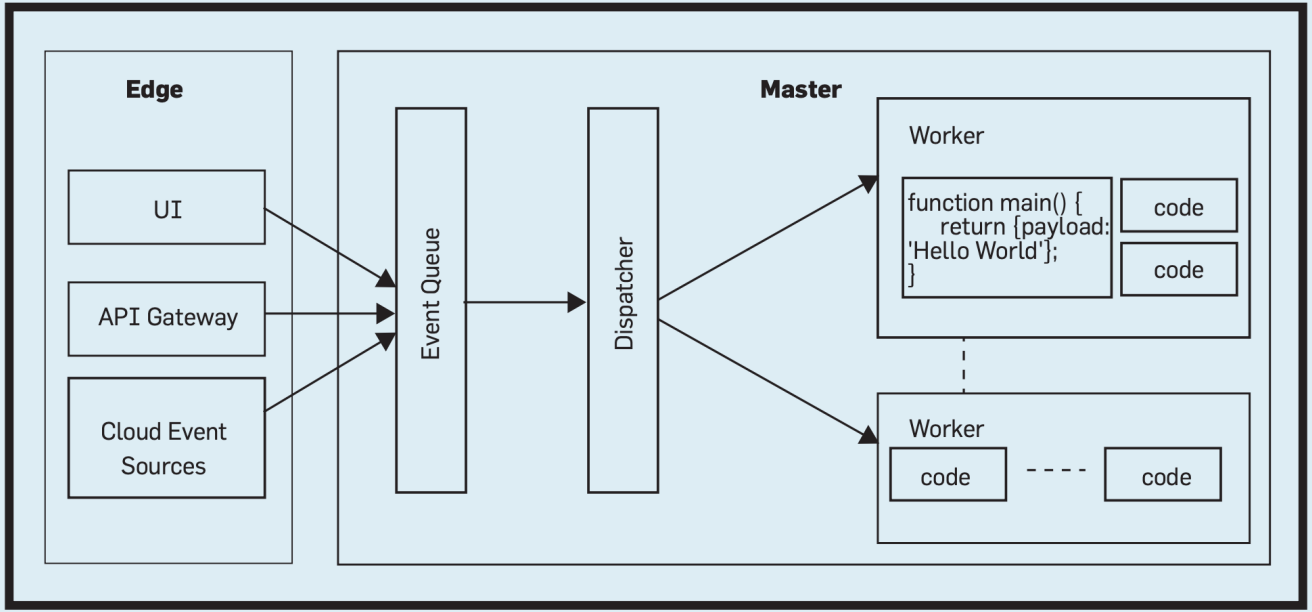
	IaaS	1st Gen PaaS	FaaS	BaaS/SaaS
Expertise required	High	Medium	Low	Low
Developer Control/Customization allowed	High	Medium	Low	Very low
Scaling/Cost	Requires high-level of expertise to build auto-scaling rules and tune them	Requires high-level of expertise to build auto-scaling rules and tune them	Auto-scaling to work load requested (function calls), and only paying for when running (scale to zero)	Hidden from users, limits set based on pricing and QoS
Unit of work deployed	Low-level infrastructure building blocks (VMs, network, storage)	Packaged code that is deployed and running as a service	One function execution	App-specific extensions
Granularity of billing	Medium to large granularity: minutes to hours per resource to years for discount pricing	Medium to large granularity: minutes to hours per resource to years for discount pricing	Very low granularity: hundreds of milliseconds of function execution time	Large: typically, subscription available based on maximum number of users and billed in months

BaaS – back-end as a service, e.g., object storage or databases

Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. (2019). The rise of serverless computing. Communications of the ACM, 62(12), 44-54.

FaaS platform architecture

Figure 1. High-level serverless FaaS platform architecture.



Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. (2019). The rise of serverless computing. Communications of the ACM, 62(12), 44-54.

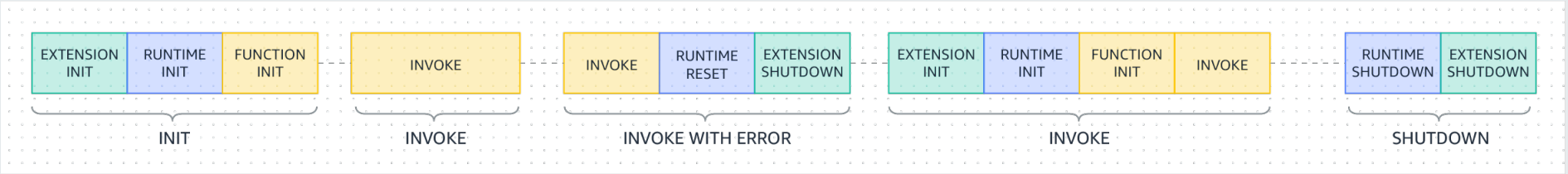
AWS Lambda functions

Function: is a resource that you can invoke to run your code in AWS Lambda. A function has code that processes events, and a runtime that passes requests and responses between Lambda and the function code. You provide the code, and you can use the provided runtimes or create your own.

- **Runtime**: Lambda runtimes allow functions in different languages to run in the same base execution environment. You configure your function to use a runtime that matches your programming language. The runtime sits in between the Lambda service and your function code, relaying invocation events, context information, and responses between the two. You can use runtimes provided by Lambda or build your own.
- **Event**: is a JSON formatted document that contains data for a function to process. The Lambda runtime converts the event to an object and passes it to your function code. When you invoke a function, you determine the structure and contents of the event. When an AWS service invokes your function, the service defines the event.
- **Concurrency**: is the number of requests that your function is serving at any given time. When your function is invoked, Lambda provisions an instance of it to process the event. When the function code finishes running, it can handle another request. If the function is invoked again while a request is still being processed, another instance is provisioned, increasing the function's concurrency.
- **Trigger**: is a resource or configuration that invokes a Lambda function. This includes AWS services that can be configured to invoke a function, applications that you develop, and event source mappings. An event source mapping is a resource in Lambda that reads items from a stream or queue and invokes a function.

AWS Lambda Concepts https://docs.aws.amazon.com/lambda/latest/dg/welcome.html, available in PDF.

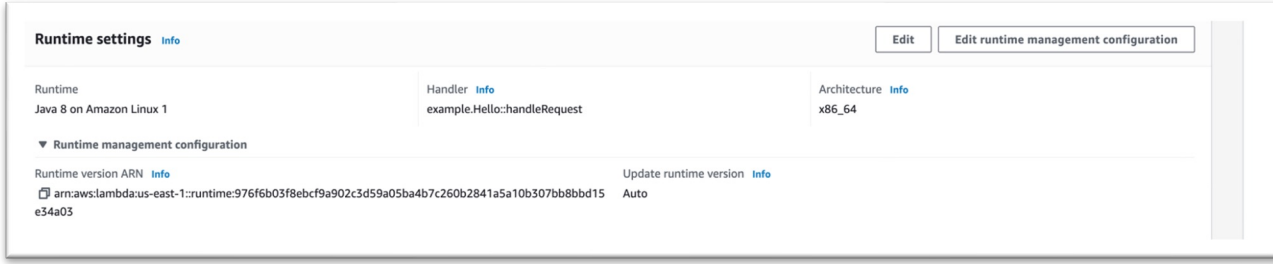
AWS Lambda functions, how does it works?



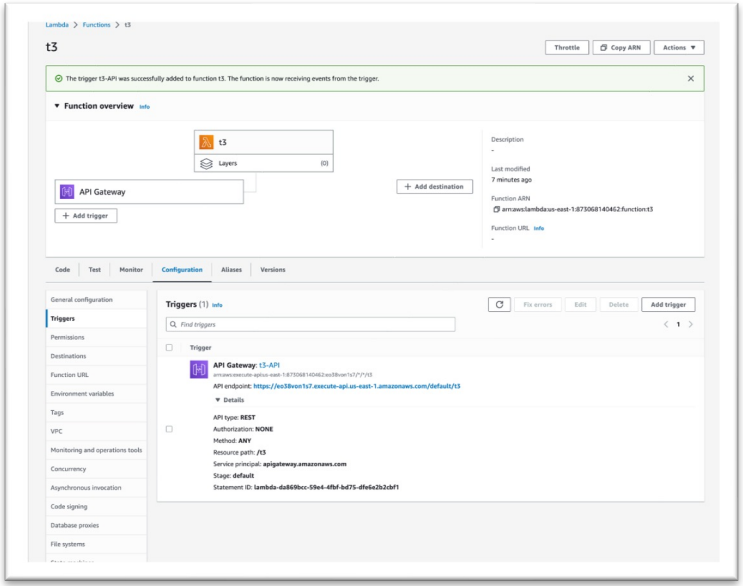
https://docs.aws.amazon.com/lambda/latest/dg/lambda-runtime-environment.html#runtimes-lifecycle-restore

AWS Lambda functions

Upload the code

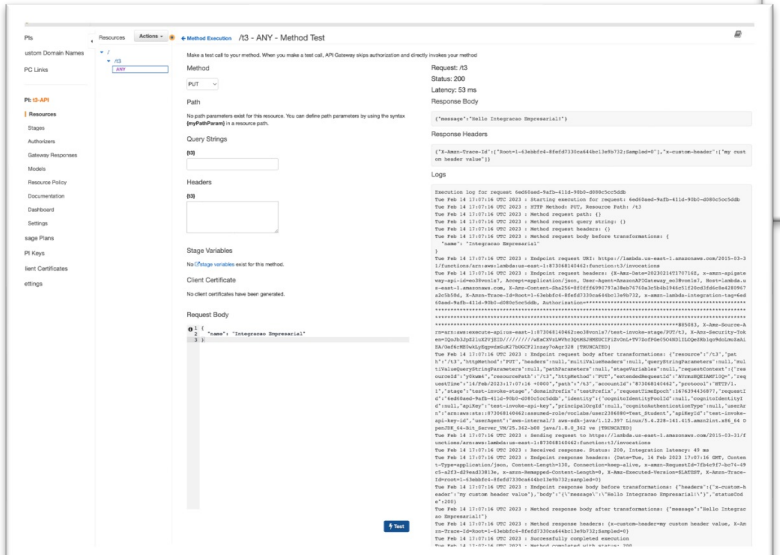


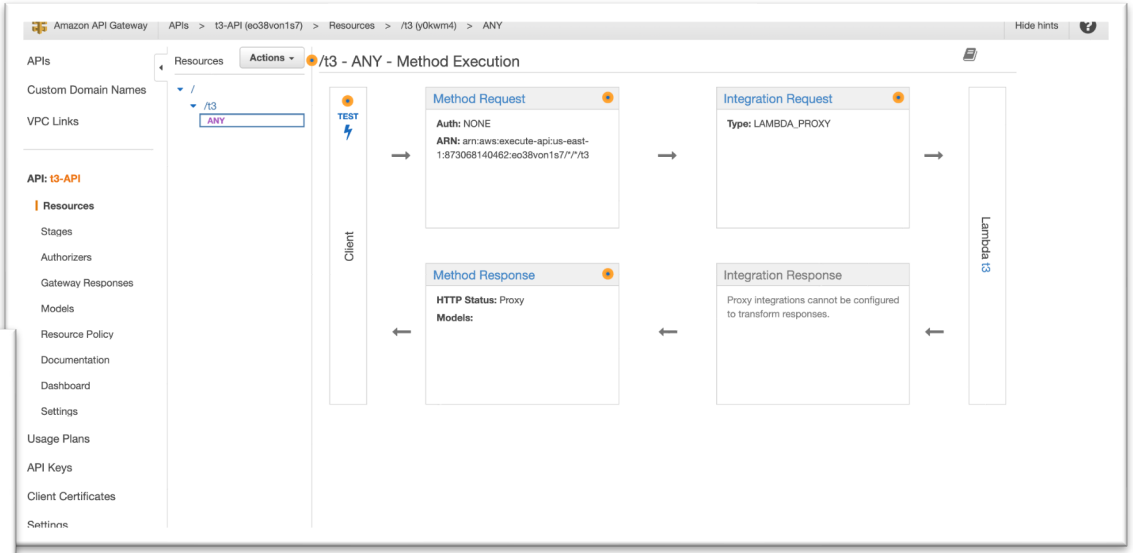
Define a trigger



AWS Lambda functions

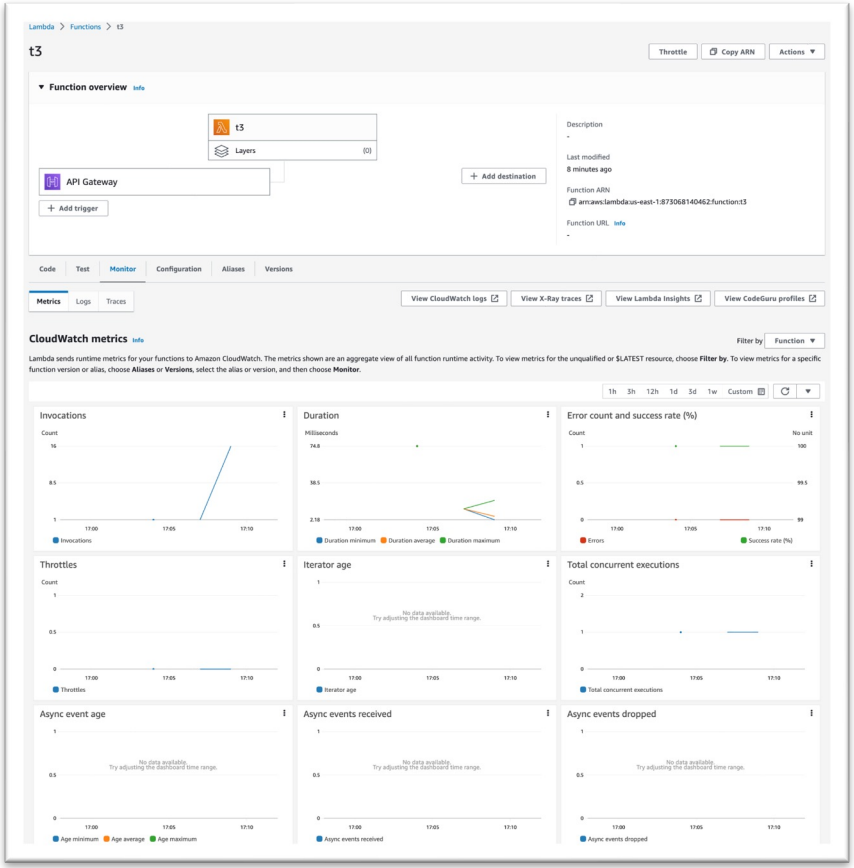
Test the function





AWS Lambda functions

Check consumption



Serverless drawbacks

- Not suitable in terms of cost and execution platform for long-running processes
- Vendor lock-in
- Cold starts can cause an overhead that is unacceptable on low-latency applications
- Monitoring and debugging is more complex when compared with other architectures

Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. (2019). The rise of serverless computing. Communications of the ACM, 62(12), 44-54.

TÉCNICO LISBOA

Software Bill of Materials (SBOM)

Bill of Materials (BOM)

LOW FAT VANILLA FLAVOURED YOGHURT

INGREDIENTS: Skim **Milk**, Concentrated Skim **Milk**, Water, Sugar, Cream (From **Milk**), Thickeners (1422 (From Maize), 1442 (From Maize)), **Milk** Solids, Gelatine, Flavours, Acidity Regulators (331, 332, 270, 330), Enzyme (Lactase), Live Cultures.

Contains Milk and Milk Products.

The Minimum Elements
For a Software Bill of Materials (SBOM)

Pursuant to
Executive Order 14028
on Improving the Nation's Cybersecurity

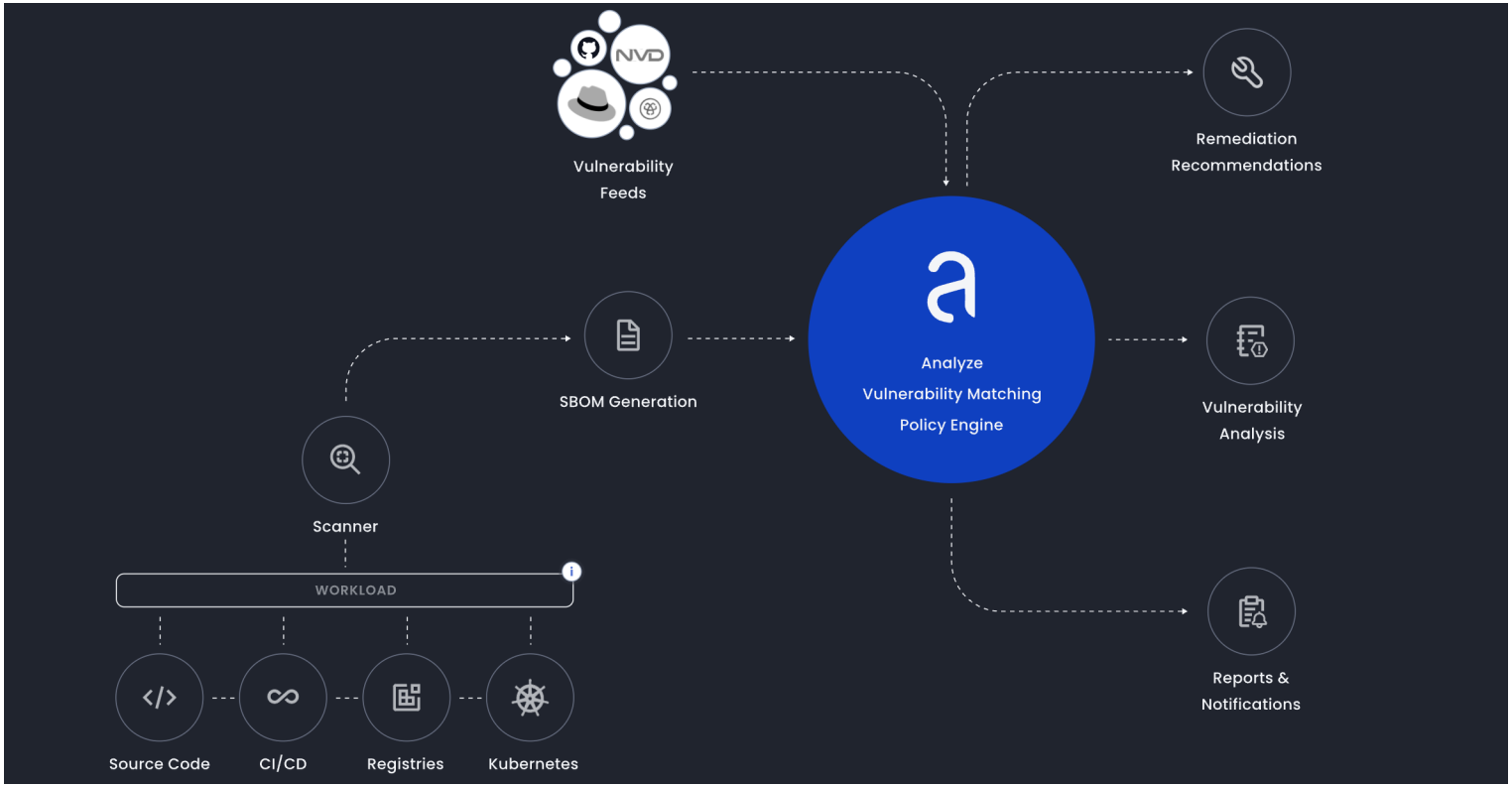
The United States Department of Commerce

July 12, 2021

SBOM

https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf

How Container Vulnerability Scanning works
https://anchore.com/container-vulnerability-scanning/



How Kubernetes Image Scanning works
https://anchore.com/kubernetes/

Preview of open-source available tools

https://anchore.com/opensource/

Syft

A CLI tool for generating a Software Bill of Materials (SBOM) from container images and filesystems.

Try Syft

Watch in action

Grype

An easy-to-integrate open source vulnerability scanning tool for container images and filesystems.

Try Grype

Watch in action

```
sergioguerreiro@kdbio45 DockerFileKafka % docker build . -t kafka
[+] Building 40.8s (26/26) FINISHED
 => [internal] load build definition from Dockerfile                                                0.0s
 => => transferring dockerfile: 923B                                                                0.0s
 => [internal] load .dockerignore                                                                   0.0s
 => => transferring context: 2B                                                                     0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                    2.4s
 => [internal] load build context                                                                   2.3s
 => => transferring context: 101.34MB                                                               2.3s
 => [ 1/21] FROM docker.io/library/ubuntu@sha256:20fa2d7bb4de7723f542be5923b06c4d704370f0390e4ae9e1c83  2.2s
 => => resolve docker.io/library/ubuntu@sha256:20fa2d7bb4de7723f542be5923b06c4d704370f0390e4ae9e1c833c  0.0s
 => => sha256:20fa2d7bb4de7723f542be5923b06c4d704370f0390e4ae9e1c833c8785644c1 1.42kB / 1.42kB       0.0s
 => => sha256:1bc0bc3815bdcfafefa6b3ef1d8fd159564693d0f8fbb37b8151074651a11ffb 529B / 529B          0.0s
 => => sha256:21735dab04ba60a144c23c491690a46073e06c72fdcc100f61bc610124eaa8c9 1.48kB / 1.48kB      0.0s
 => => sha256:00f50047d6061c27e70588a5aab89adada756e87d782a6c6bd08b4139eb8ea10 28.38MB / 28.38MB    0.7s
 => => extracting sha256:00f50047d6061c27e70588a5aab89adada756e87d782a6c6bd08b4139eb8ea10            1.4s
 => [ 2/21] RUN apt-get update                                                                      2.3s
 => [ 3/21] RUN apt-get install -y sudo                                                             1.3s
 => [ 4/21] RUN sudo apt-get install -y openjdk-8-jdk                                              19.0s
 => [ 5/21] RUN apt-get install -y telnet                                                           1.1s
 => [ 6/21] RUN apt-get install -y vim                                                              2.2s
 => [ 7/21] RUN apt-get install -y git                                                              2.7s
 => [ 8/21] RUN apt-get install -y maven                                                            2.9s
 => [ 9/21] COPY apache-zookeeper-3.8.0-bin.tar.gz /root                                            0.0s
 => [10/21] COPY kafka_2.13-3.1.0.tgz /root                                                         0.1s
 => [11/21] COPY zoo.cfg /root                                                                      0.0s
 => [12/21] COPY start.sh /root                                                                     0.0s
 => [13/21] RUN tar -zxf /root/apache-zookeeper-3.8.0-bin.tar.gz                                    0.4s
 => [14/21] RUN mv apache-zookeeper-3.8.0-bin /usr/local/zookeeper                                  0.4s
 => [15/21] RUN mkdir -p /var/lib/zookeeper                                                         0.2s
 => [16/21] RUN mv /root/zoo.cfg /usr/local/zookeeper/conf                                          0.1s
 => [17/21] RUN tar -zxf /root/kafka_2.13-3.1.0.tgz                                                 0.7s
 => [18/21] RUN mv kafka_2.13-3.1.0 /usr/local/kafka                                                0.3s
 => [19/21] RUN mkdir /tmp/kafka-logs                                                               0.2s
 => [20/21] RUN echo "/usr/local/zookeeper/bin/zkServer.sh start" >> /root/.profile                 0.3s
 => [21/21] RUN echo "sudo /usr/local/kafka/bin/kafka-server-start.sh -daemon /usr/local/kafka/config/  0.2s
 => exporting to image                                                                              1.8s
 => => exporting layers                                                                             1.8s
 => => writing image sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6        0.0s
 => => naming to docker.io/library/kafka                                                            0.0s
```

```
% ./syft packages sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6 -o table >> output
✔ Loaded image
✔ Parsed image
✔ Cataloged packages      [637 packages]
```

Human readable format

NAME	VERSION	TYPE
PyGObject	3.42.1	python
US_export_policy		java-archive
activation	1.1.1	java-archive
adduser	3.118ubuntu5	deb
adwaita-icon-theme	41.0-1ubuntu1	deb
alsa-topology-conf	1.2.5.1-2	deb
alsa-ucm-conf	1.2.6.3-1ubuntu1	deb
aopalliance	1.0	java-archive
aopalliance-repackaged	2.6.1	java-archive
apt	2.4.7	deb
argparse4j	0.7.0	java-archive
at-spi2-core	2.44.0-3	deb
atinject-jsr330-api	1.0	java-archive
audience-annotations	0.12.0	java-archive
audience-annotations	0.5.0	java-archive
base-files	12ubuntu4.2	deb
base-passwd	3.5.52build1	deb
bash	5.1-6ubuntu1	deb
bsdutils	1:2.37.2-4ubuntu3	deb
ca-certificates	20211016	deb
ca-certificates-java		java-archive
ca-certificates-java	20190909	deb
cdi-api	1.2	java-archive
cglib	3.2.12	java-archive
charsets		java-archive
cldrdata		java-archive
commons-cli	1.4	java-archive
commons-codec	1.14	java-archive
commons-io	2.11.0	java-archive
commons-io	2.6	java-archive
commons-lang3	3.11	java-archive
commons-lang3	3.8.1	java-archive
connect-api	3.1.0	java-archive
connect-basic-auth-extension	3.1.0	java-archive
connect-file	3.1.0	java-archive
connect-json	3.1.0	java-archive
connect-mirror	3.1.0	java-archive
connect-mirror-client	3.1.0	java-archive
connect-runtime	3.1.0	java-archive
connect-transforms	3.1.0	java-archive
coreutils	8.32-4.1ubuntu1	deb
dash	0.5.11+git20210903+057cd650a4ed-3build1	deb
dbus	1.12.20-2ubuntu4	deb
dbus-python	1.2.18	python
dbus-user-session	1.12.20-2ubuntu4	deb
dconf-gsettings-backend	0.40.0-3	deb
dconf-service	0.40.0-3	deb
debconf	1.5.79ubuntu1	deb
debianutils	5.5-1ubuntu2	deb
diffutils	1:3.8-0ubuntu2	deb
dmsetup	2:1.02.175-2.1ubuntu4	deb
dnsns		java-archive
dpkg	1.21.1ubuntu2.1	deb
dt		java-archive
e2fsprogs	1.46.5-2ubuntu1.1	deb

TÉCNICO LISBOA

JSON format

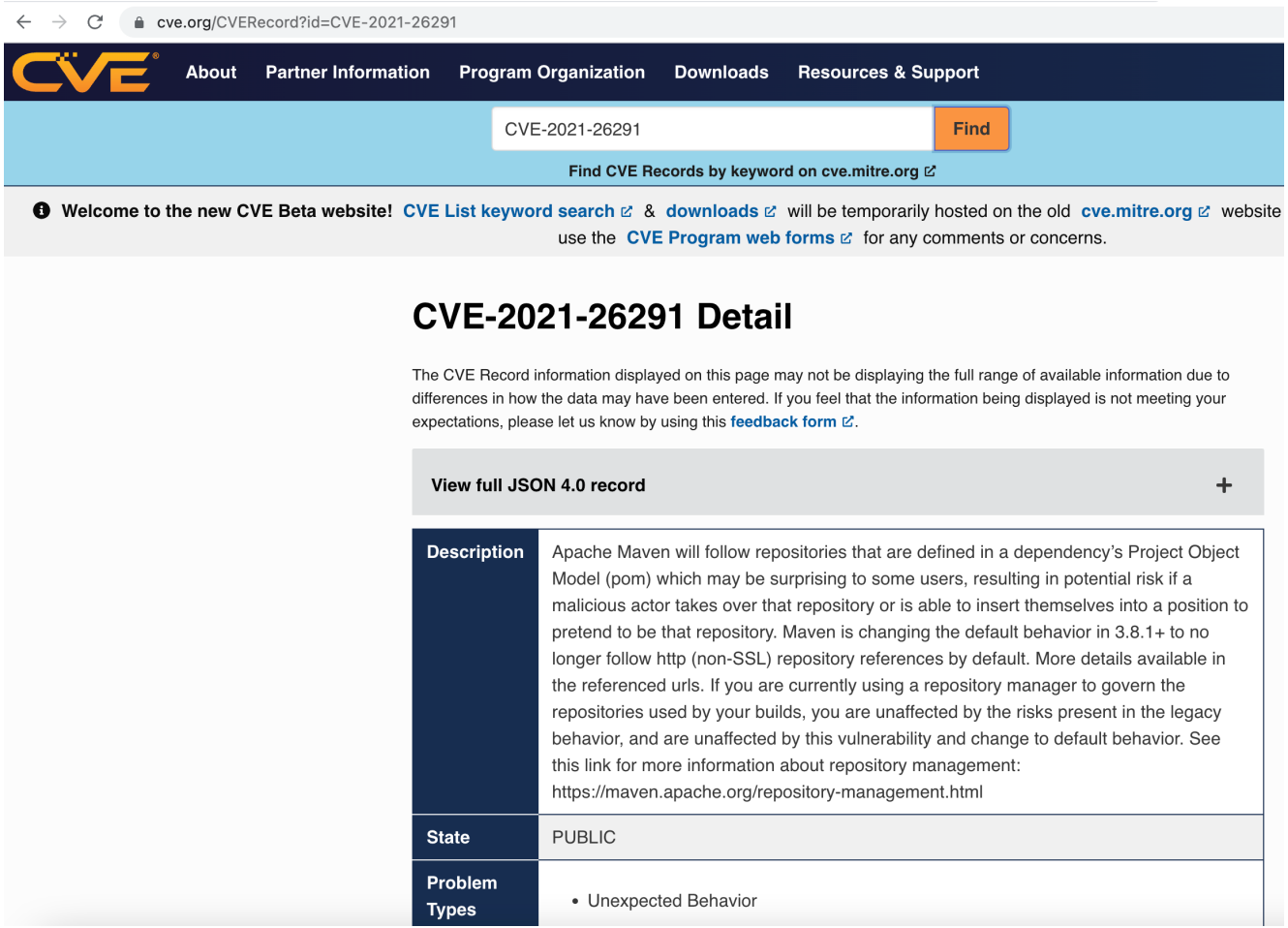
syft — more output — 163×60


```
{
  "version": 0,
  "job": {},
  "detector": {
    "name": "syft",
    "url": "https://github.com/anchore/syft",
    "version": "0.56.0"
  },
  "metadata": {
    "syft:distro": "pkg:generic/ubuntu@22.04?like=debian"
  },
  "manifests": {
    "sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6:/lib/jvm/java-1.8.0-openjdk-arm64/jre/lib/charsets.jar": {
      "name": "sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6:/lib/jvm/java-1.8.0-openjdk-arm64/jre/lib/charsets.jar",
      "file": {
        "source_location": "sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6:/lib/jvm/java-1.8.0-openjdk-arm64/jre/lib/charsets.jar"
      },
      "metadata": {
        "syft:filesystem": "sha256:cd02f67aba6a5a352784d578a08e7cfbee45571907658fd1ce9b5d29d36ea3f6"
      },
      "resolved": {
        "pkg:maven/charsets/charsets": {
          "package_url": "pkg:maven/charsets/charsets",
          "relationship": "direct",
          "scope": "runtime"
        }
      }
    },
    "sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6:/lib/jvm/java-1.8.0-openjdk-arm64/jre/lib/ext/cldrdata.jar": {
      "name": "sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6:/lib/jvm/java-1.8.0-openjdk-arm64/jre/lib/ext/cldrdata.jar",
      "file": {
        "source_location": "sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6:/lib/jvm/java-1.8.0-openjdk-arm64/jre/lib/ext/cldrdata.jar"
      },
      "metadata": {
        "syft:filesystem": "sha256:cd02f67aba6a5a352784d578a08e7cfbee45571907658fd1ce9b5d29d36ea3f6"
      },
      "resolved": {
        "pkg:maven/cldrdata/cldrdata": {
          "package_url": "pkg:maven/cldrdata/cldrdata",
          "relationship": "direct",
          "scope": "runtime"
        }
      }
    },
    "sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6:/lib/jvm/java-1.8.0-openjdk-arm64/jre/lib/ext/dnsns.jar": {
      "name": "sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6:/lib/jvm/java-1.8.0-openjdk-arm64/jre/lib/ext/dnsns.jar",
      "file": {
        "source_location": "sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6:/lib/jvm/java-1.8.0-openjdk-arm64/jre/lib/ext/dnsns.jar"
      },
      "metadata": {
        "syft:filesystem": "sha256:cd02f67aba6a5a352784d578a08e7cfbee45571907658fd1ce9b5d29d36ea3f6"
      },
      "resolved": {
        "pkg:maven/dnsns/dnsns": {
          "package_url": "pkg:maven/dnsns/dnsns",
          "relationship": "direct",
          "scope": "runtime"
        }
      }
output
```

TÉCNICO LISBOA

```
% ./grype sha256:ce774c2bb577d868ec312839d63b9f4545ec20027a69dffc2f78e2fadfaaf7b6
✔ Vulnerability DB        [updated]
✔ Loaded image
✔ Parsed image
✔ Cataloged packages      [637 packages]
✔ Scanned image           [444 vulnerabilities]
```

NAME	INSTALLED	FIXED-IN	TYPE	VULNERABILITY	SEVERITY
commons-io	2.6		java-archive	CVE-2021-29425	Medium
commons-io	2.6	2.7	java-archive	GHSA-gwrp-pvrq-jmwv	Medium
coreutils	8.32-4.1ubuntu1		deb	CVE-2016-2781	Low
geronimo-annotation_1.3_spec	1.3		java-archive	CVE-2008-0732	Low
geronimo-annotation_1.3_spec	1.3		java-archive	CVE-2011-5034	High
geronimo-interceptor_3.0_spec	1.0.1		java-archive	CVE-2008-0732	Low
geronimo-interceptor_3.0_spec	1.0.1		java-archive	CVE-2011-5034	High
git	1:2.34.1-1ubuntu1.4		deb	CVE-2018-1000021	Low
git-man	1:2.34.1-1ubuntu1.4		deb	CVE-2018-1000021	Low
guava	29.0-jre		java-archive	CVE-2020-8908	Low
guava	29.0-jre		java-archive	GHSA-5mg8-w23w-74h3	Low
httpclient	4.5.11		java-archive	CVE-2020-13956	Medium
httpclient	4.5.11	4.5.13	java-archive	GHSA-7r82-7xv7-xcpj	Medium
jackson-databind	2.12.3	2.12.6.1	java-archive	GHSA-57j2-w4cx-62h2	High
jackson-databind	2.13.1	2.13.2.1	java-archive	GHSA-57j2-w4cx-62h2	High
jackson-databind	2.12.3		java-archive	CVE-2020-36518	High
jackson-databind	2.13.1		java-archive	CVE-2020-36518	High
jetty-client	9.4.43.v20210629		java-archive	CVE-2022-2047	Low
jetty-client	9.4.43.v20210629		java-archive	CVE-2022-2048	High
jetty-continuation	9.4.43.v20210629		java-archive	CVE-2022-2048	High
jetty-continuation	9.4.43.v20210629		java-archive	CVE-2022-2047	Low
jetty-http	9.4.43.v20210629		java-archive	CVE-2022-2047	Low
jetty-http	9.4.43.v20210629	9.4.47	java-archive	GHSA-cj7v-27pg-wf7q	Low
jetty-http	9.4.43.v20210629		java-archive	CVE-2022-2048	High
jetty-io	9.4.43.v20210629		java-archive	CVE-2022-2048	High
jetty-io	9.4.43.v20210629		java-archive	CVE-2022-2047	Low
jetty-security	9.4.43.v20210629		java-archive	CVE-2022-2047	Low
jetty-security	9.4.43.v20210629		java-archive	CVE-2022-2048	High
jetty-server	9.4.43.v20210629		java-archive	CVE-2022-2047	Low
jetty-server	9.4.43.v20210629		java-archive	CVE-2022-2048	High
jetty-servlet	9.4.43.v20210629		java-archive	CVE-2022-2047	Low
jetty-servlet	9.4.43.v20210629		java-archive	CVE-2022-2048	High
jetty-servlets	9.4.43.v20210629		java-archive	CVE-2022-2047	Low
jetty-servlets	9.4.43.v20210629		java-archive	CVE-2022-2048	High
jetty-util	9.4.43.v20210629		java-archive	CVE-2022-2047	Low
jetty-util	9.4.43.v20210629		java-archive	CVE-2022-2048	High
jetty-util-ajax	9.4.43.v20210629		java-archive	CVE-2022-2048	High
jetty-util-ajax	9.4.43.v20210629		java-archive	CVE-2022-2047	Low
jsoup	1.10.2	1.14.2	java-archive	GHSA-m72m-mhq2-9p6c	High
jsoup	1.10.2		java-archive	CVE-2022-36033	Medium
jsoup	1.10.2	1.15.3	java-archive	GHSA-gp7f-rwcx-9369	Medium
jsoup	1.10.2		java-archive	CVE-2021-37714	High
libapparmor1	3.0.4-2ubuntu2.1		deb	CVE-2016-1585	Medium
libc-bin	2.35-0ubuntu3.1		deb	CVE-2016-20013	Negligible
libc6	2.35-0ubuntu3.1		deb	CVE-2016-20013	Negligible
libcairo-gobject2	1.16.0-5ubuntu2		deb	CVE-2019-6461	Low
libcairo-gobject2	1.16.0-5ubuntu2		deb	CVE-2018-18064	Low
libcairo-gobject2	1.16.0-5ubuntu2		deb	CVE-2017-7475	Low
libcairo2	1.16.0-5ubuntu2		deb	CVE-2019-6461	Low
libcairo2	1.16.0-5ubuntu2		deb	CVE-2018-18064	Low
libcairo2	1.16.0-5ubuntu2		deb	CVE-2017-7475	Low



cve.org/CVERecord?id=CVE-2021-26291

About | Partner Information | Program Organization | Downloads | Resources & Support

CVE-2021-26291 Find

Find CVE Records by keyword on cve.mitre.org

Welcome to the new CVE Beta website! CVE List keyword search & downloads will be temporarily hosted on the old cve.mitre.org website use the CVE Program web forms for any comments or concerns.

CVE-2021-26291 Detail

The CVE Record information displayed on this page may not be displaying the full range of available information due to differences in how the data may have been entered. If you feel that the information being displayed is not meeting your expectations, please let us know by using this feedback form.

View full JSON 4.0 record +

Description	Apache Maven will follow repositories that are defined in a dependency's Project Object Model (pom) which may be surprising to some users, resulting in potential risk if a malicious actor takes over that repository or is able to insert themselves into a position to pretend to be that repository. Maven is changing the default behavior in 3.8.1+ to no longer follow http (non-SSL) repository references by default. More details available in the referenced urls. If you are currently using a repository manager to govern the repositories used by your builds, you are unaffected by the risks present in the legacy behavior, and are unaffected by this vulnerability and change to default behavior. See this link for more information about repository management: https://maven.apache.org/repository-management.html
State	PUBLIC
Problem Types	• Unexpected Behavior

TÉCNICO LISBOA

Q&A

TÉCNICO LISBOA