

SUPSI

Introduction to Dev Containers

Massimo Coluzzi

Content realized in collaboration with:
T. Leidi, R. Guidi

Big Data Processing
Bachelor in Data Science

Docker

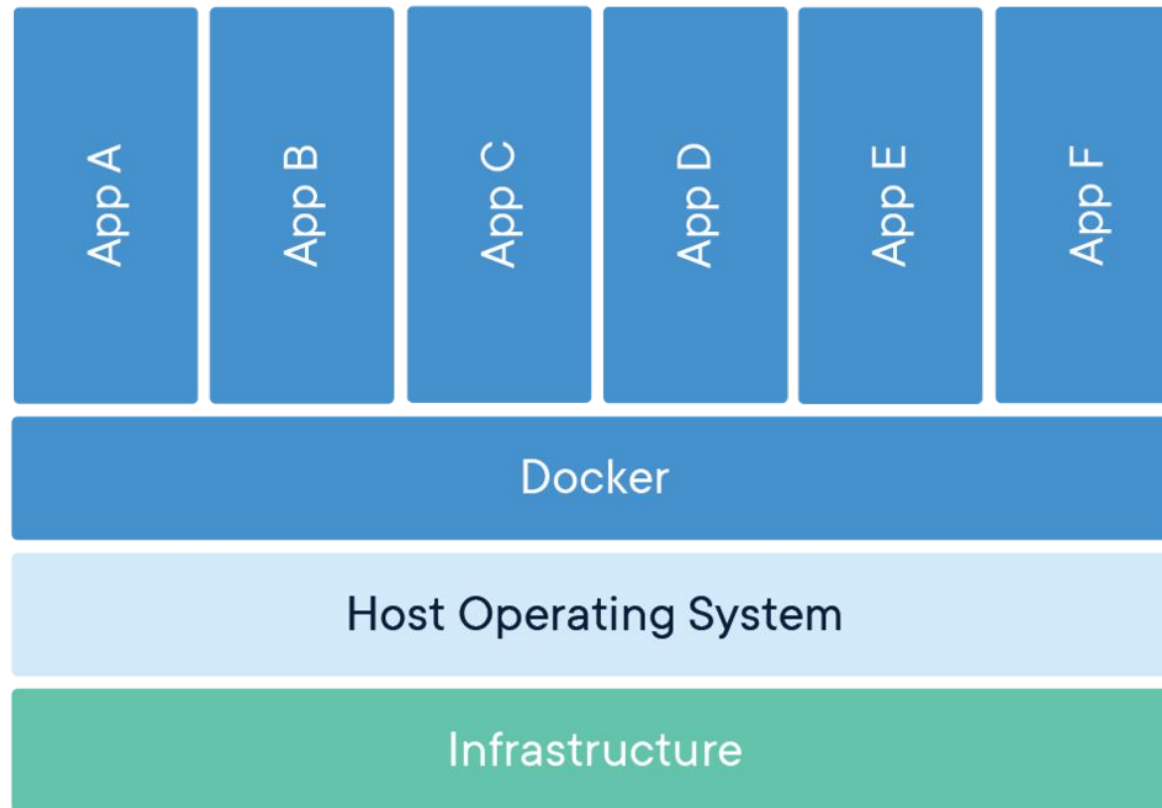
Docker is a set of tools and an execution environment to perform **operating-system-level virtualization**.

Docker is used to build, manage and run software packages called **"containers"** with a universal packaging approach that bundles up application **executables and dependencies** together.

Containers isolate software from its environment and ensure that it works uniformly, **regardless of the infrastructure**.

Docker is available both for Linux and Windows-based applications.

Docker



Why Docker

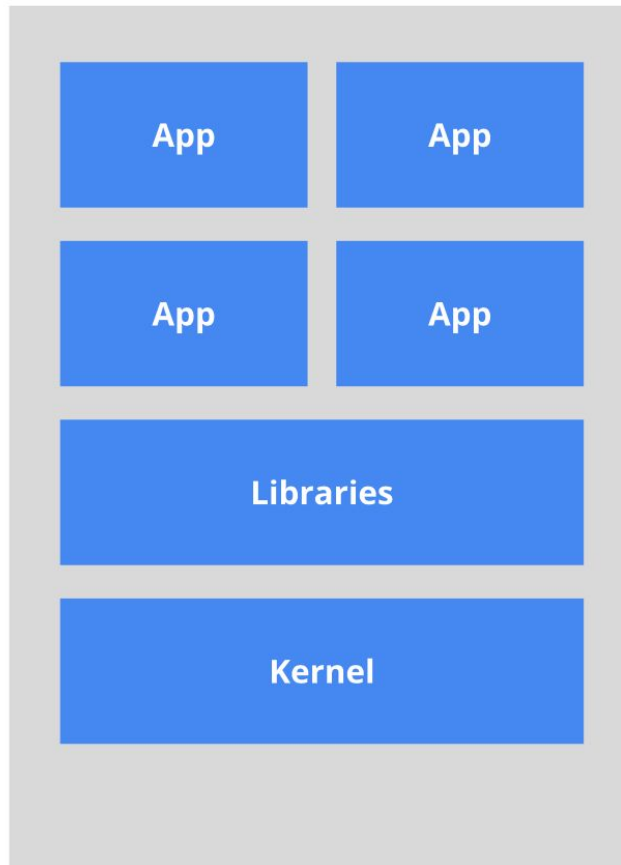
With Docker it is possible to run a **command isolated from everything else** on the system.

It will only access exactly the resources it is allowed to (storage, CPU, memory), and won't know there is anything else on the machine.

The process running inside a container thinks **it's the only one**.

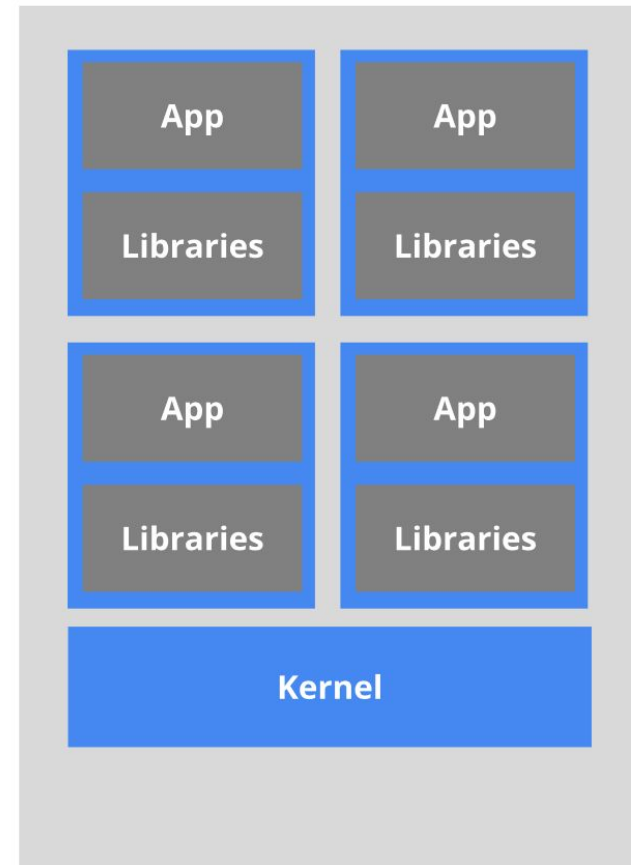
Why Docker

The old way: Applications on host



*Heavyweight, non-portable
Relies on OS package manager*

The new way: Deploy containers



*Small and fast, portable
Uses OS-level virtualization*

Containers

Containers are **more lightweight** than virtual machines.

Multiple containers can run on the same machine and share the OS kernel, each running as **isolated processes** in user space.

A container is managed using the **Docker API or CLI**.

Windows Containers

Docker and Microsoft have a joint engineering relationship to deliver a consistent Docker experience for Windows users.

All Windows Server 2016 and later versions come with Docker Engine-Enterprise.

Additionally, developers can leverage Docker natively on Windows 10 with **Hyper-V Containers**, to build, ship and run Windows Containers.

Images

Containers are instantiated from images: **lightweight, standalone, executable package of software**. Everything needed to run an application is included: code, runtime, system tools, system libraries and settings.

A Docker image is a **read-only template**. Images are often created by combining and modifying standard images downloaded from repositories.

Images are typically small: tens of MBs in size.

Architecture

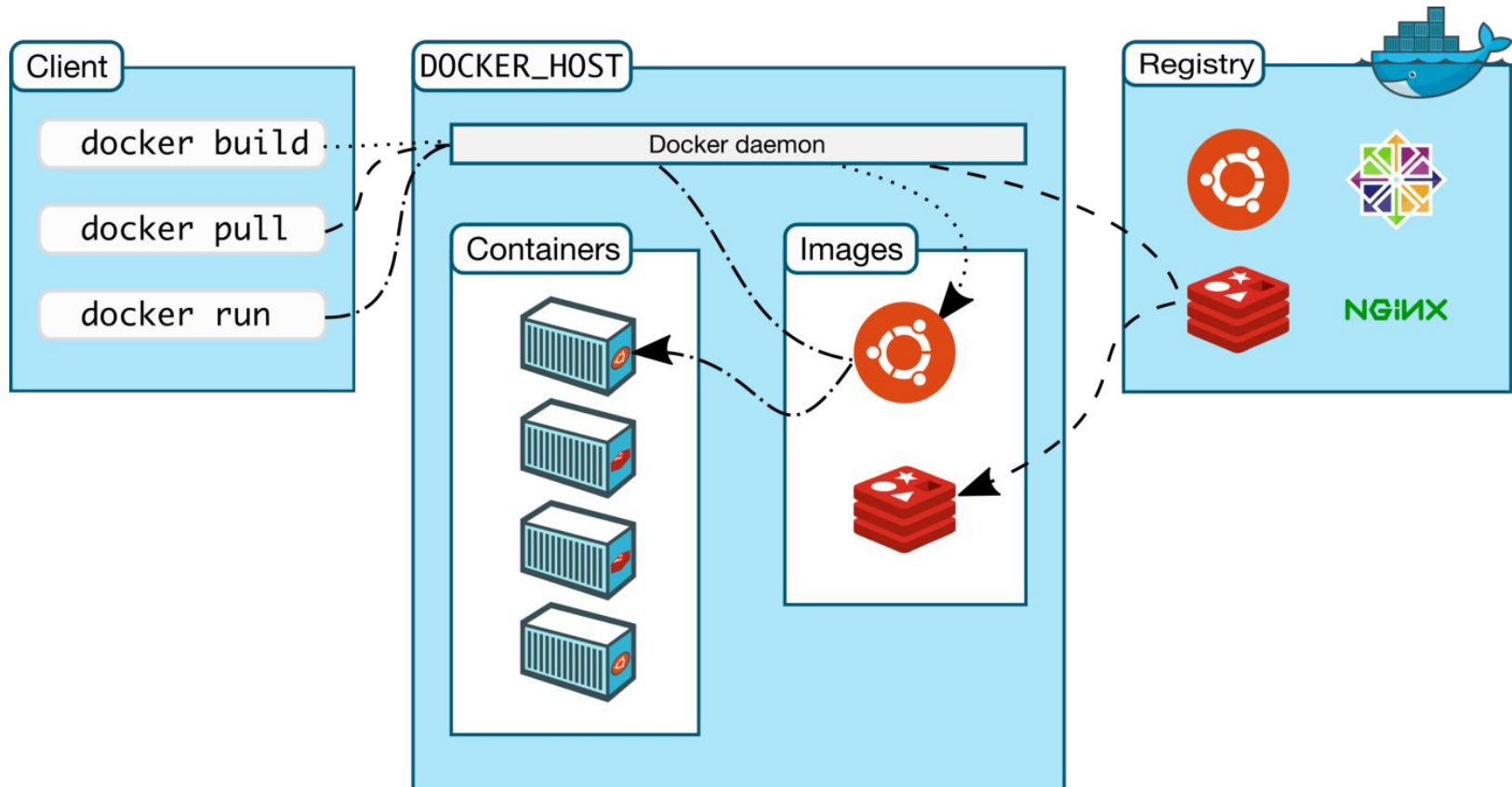
Docker is based on a **client-server architecture**.

The Docker client talks to the Docker daemon, which does the heavy lifting of building, running, and distributing your Docker containers.

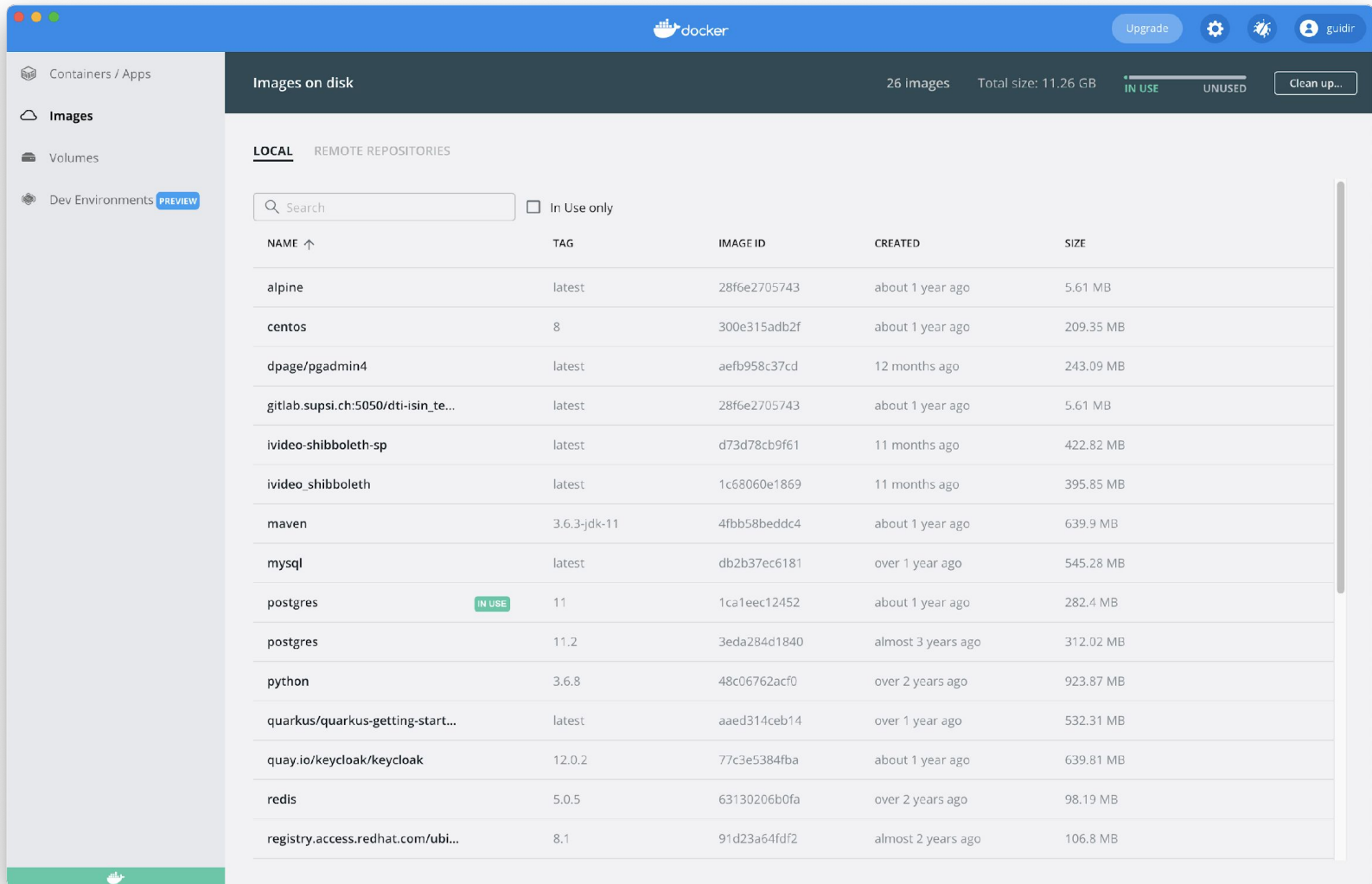
The Docker daemon, called **dockerd** is a persistent process.

The **Docker client and daemon** can run on the same system, or you can connect a Docker client to a remote Docker daemon. The Docker client and daemon communicate using a REST API, over UNIX sockets or a network interface.

Architecture



Docker Dashboard



The screenshot displays the Docker Dashboard interface. The left sidebar contains navigation links for 'Containers / Apps', 'Images', 'Volumes', and 'Dev Environments' (marked as a preview). The main content area is titled 'Images on disk' and shows a summary of 26 images with a total size of 11.26 GB. A progress bar indicates that 1 image is 'IN USE' and 25 are 'UNUSED'. A 'Clean up...' button is available. Below the summary, there are tabs for 'LOCAL' and 'REMOTE REPOSITORIES'. The 'LOCAL' tab is active, showing a search bar and a checkbox for 'In Use only'. A table lists the local images with columns for NAME, TAG, IMAGE ID, CREATED, and SIZE. The 'postgres' image is highlighted with an 'IN USE' badge.

NAME ↑	TAG	IMAGE ID	CREATED	SIZE
alpine	latest	28f6e2705743	about 1 year ago	5.61 MB
centos	8	300e315adb2f	about 1 year ago	209.35 MB
dpage/pgadmin4	latest	aefb958c37cd	12 months ago	243.09 MB
gitlab.supsi.ch:5050/dti-isin_te...	latest	28f6e2705743	about 1 year ago	5.61 MB
ivideo-shibboleth-sp	latest	d73d78cb9f61	11 months ago	422.82 MB
ivideo_shibboleth	latest	1c68060e1869	11 months ago	395.85 MB
maven	3.6.3-jdk-11	4fbb58beddc4	about 1 year ago	639.9 MB
mysql	latest	db2b37ec6181	over 1 year ago	545.28 MB
postgres	11	1ca1eec12452	about 1 year ago	282.4 MB
postgres	11.2	3eda284d1840	almost 3 years ago	312.02 MB
python	3.6.8	48c06762acf0	over 2 years ago	923.87 MB
quarkus/quarkus-getting-start...	latest	aaed314ceb14	over 1 year ago	532.31 MB
quay.io/keycloak/keycloak	12.0.2	77c3e5384fba	about 1 year ago	639.81 MB
redis	5.0.5	63130206b0fa	over 2 years ago	98.19 MB
registry.access.redhat.com/ubi...	8.1	91d23a64fdf2	almost 2 years ago	106.8 MB

Mount Points and Volumes

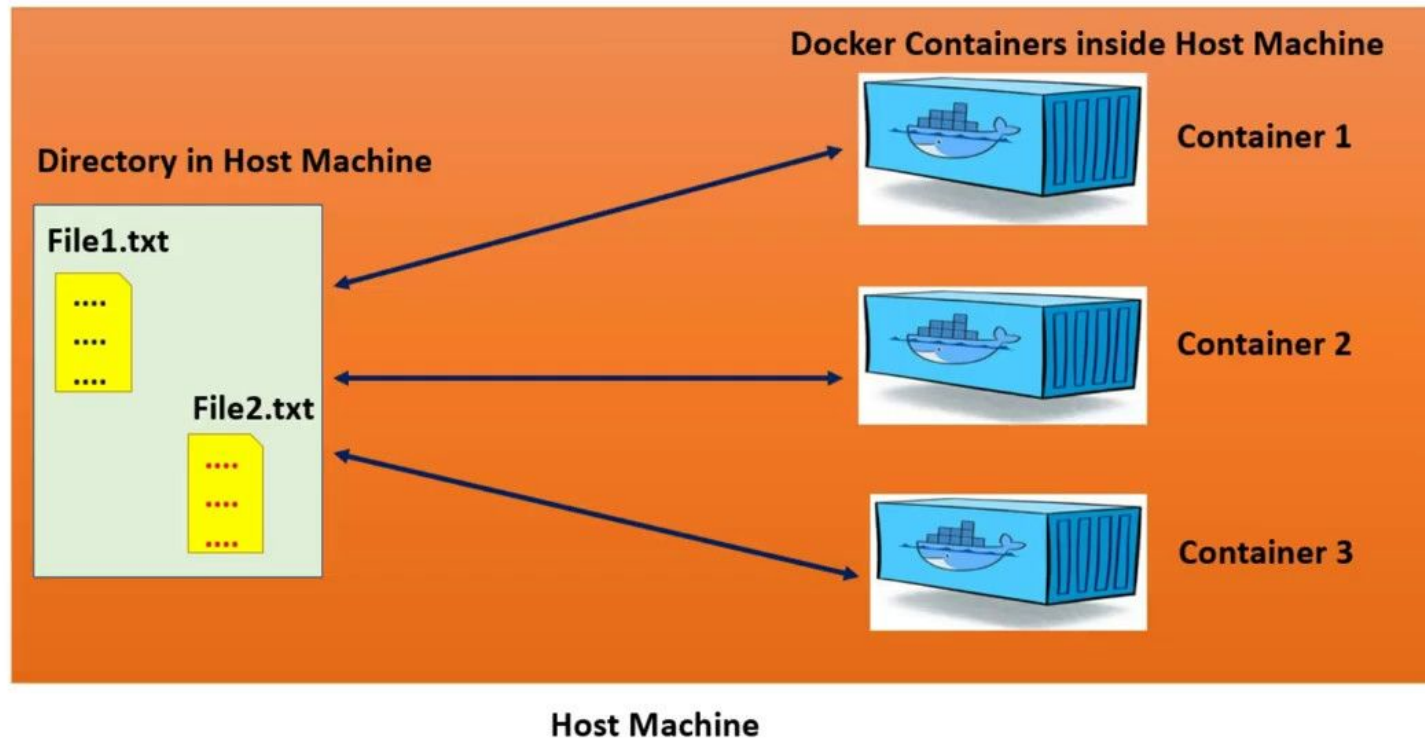
Containers are ephemeral, they leave nothing behind by default. Any changes made to a container, given that you don't save it as an image, are lost as soon as it is removed.

But having data persist is useful. That's where volumes come in.

When starting a Docker container, you can specify that certain directories are mount points for either local directories (of the host machine), or for volumes.

Mount Points and Volumes

Volumes



Dev Container

A Dev Container is a modern development technique that allows you to **use a container as a full-featured development environment**.

It allows you to **open any folder inside a container** and develop in an isolated environment.

Nowadays, many IDE supports the Dev Container development model but in this course we will rely on **Visual Studio Code**.

Dev Container

The Dev Container solution allows us to **practice the parallel programming principles** without disrupting the Python installation on your laptops.

Moreover, it allows all of you to work in the same environment, avoiding issues related to local configurations.

We will **work** on VSCode **inside a container** running the **no-GIL version of Python 3.13**.

Python 3.13 no-GIL

As we will see in more detail in the following lectures, Python's **Global Interpreter Lock** (GIL) prevents Python programs from running threads in parallel.

Python 3.13 comes with the possibility of **removing the GIL**. To enable such a feature, **Python 3.13 must be built from sources**.

The dev-container mode will **build the sources on your behalf** and prevent you from disrupting your environment.