Guardium Administration

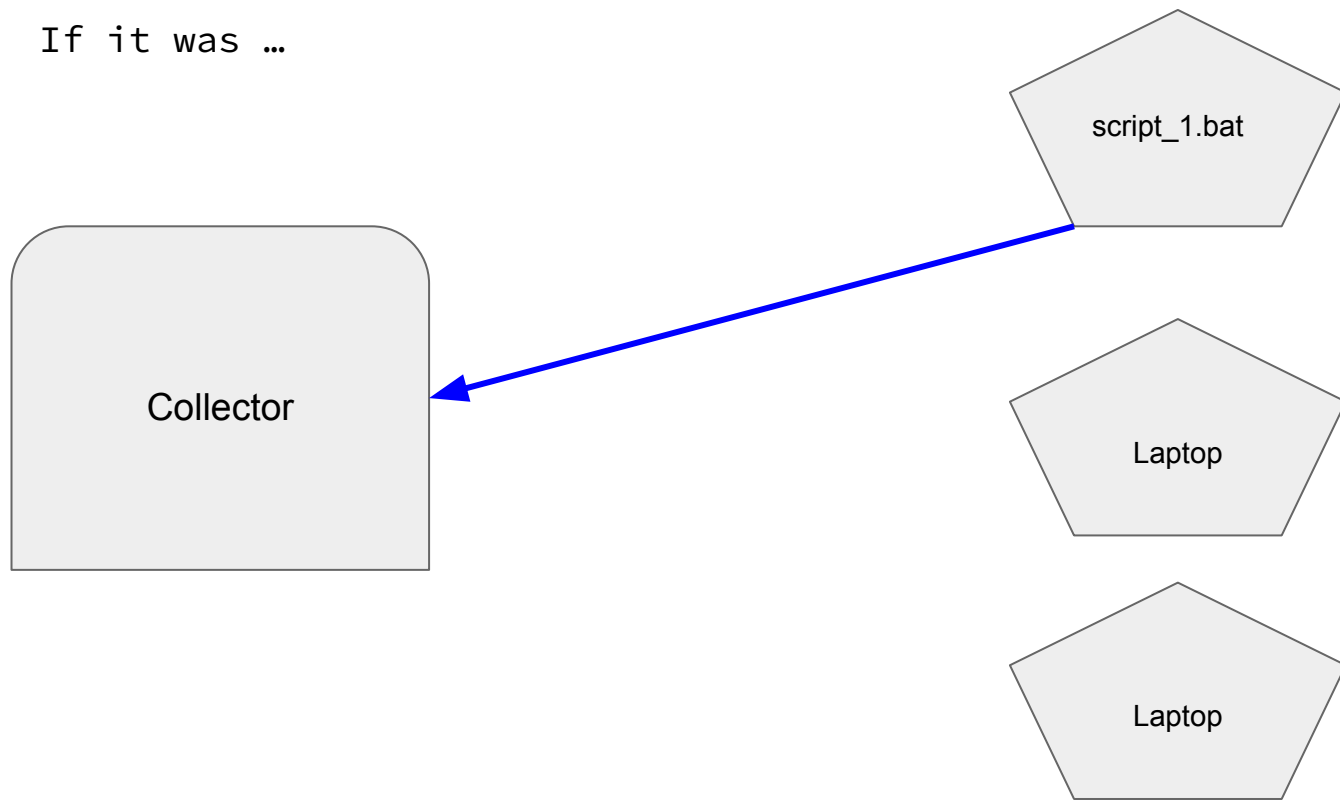## Example of an Admin Set-up for guardium

The Need for a Strong Client Side

- Guardium is a Client-Server architecture based product. However, Guardium provides only the Web interface as client.
- All the Client non web-based interface (for grdapi & cli) must be developed
- As first of all, you need a place to execute them from , and it CANNOT be your laptop

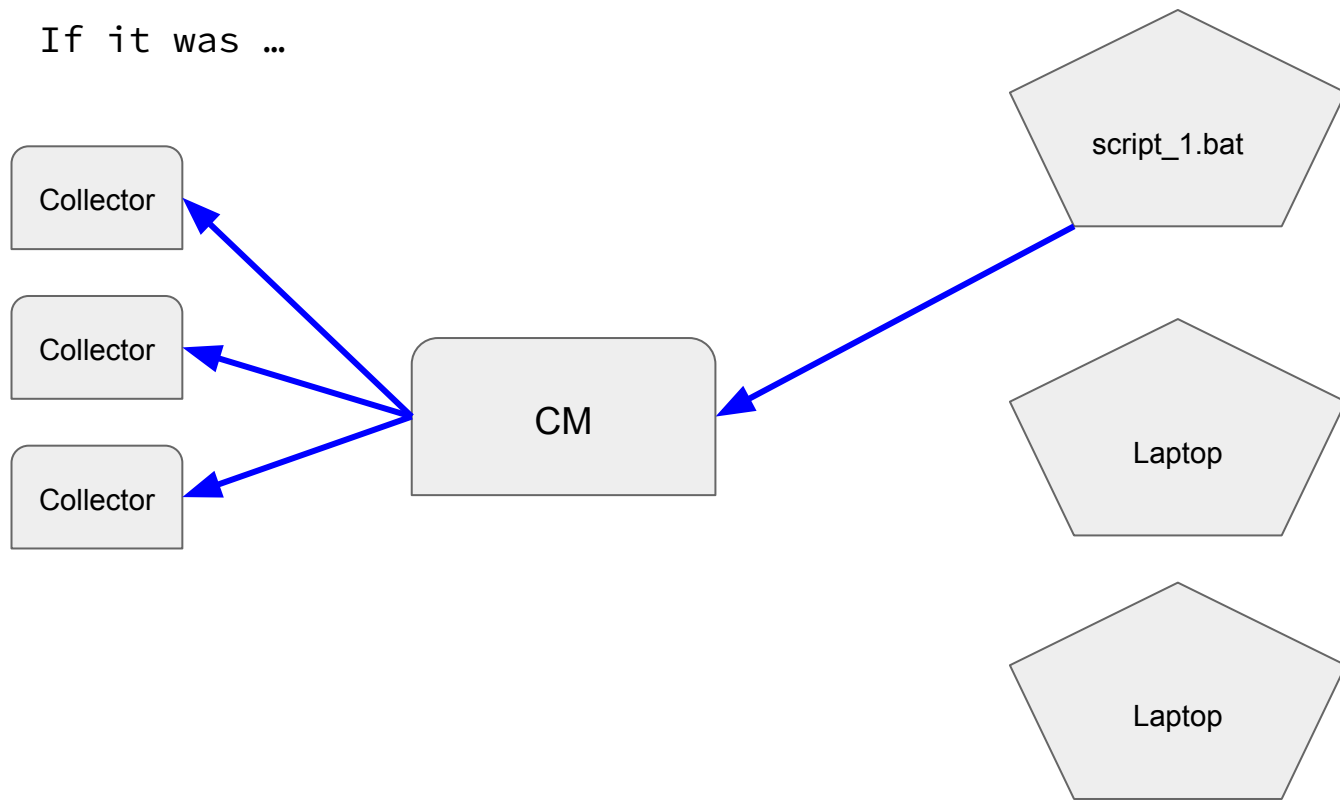# Why the client-side for grdapi Cannot be on Laptops

If it was …

script_1.bat

Collector

How do you make script_1.bat available to others than yourself ?

Laptop

Answer :
You don't …

Laptop

Why the client-side for grdapi Cannot be on Laptops

If it was …


```
Collector
```

```
Collector
```

```
Collector
```

```
CM
```

```
script_1.bat
```
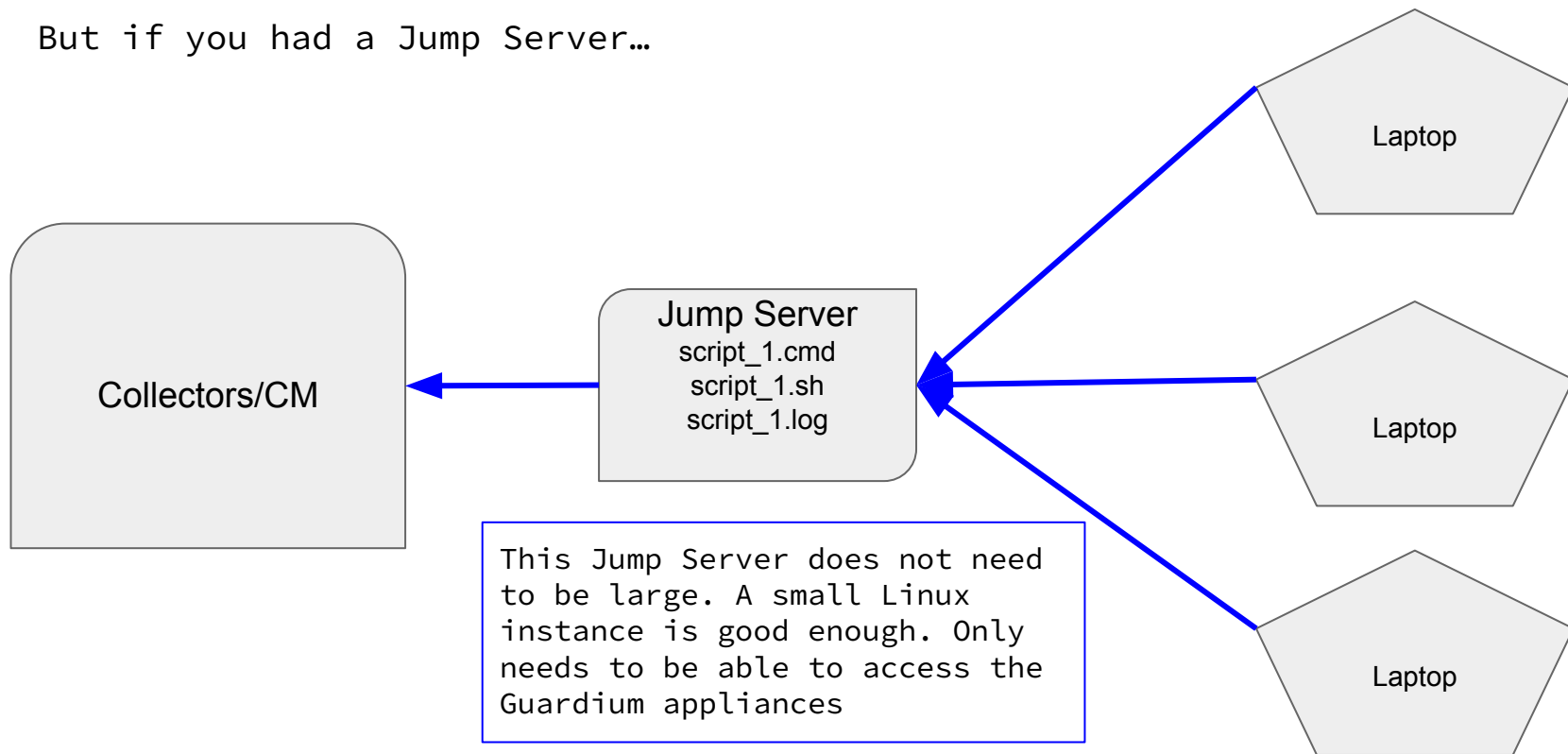
```
Laptop
```

```
Laptop
```

How do you make script_1.bat available to others than yourself ?

Answer :
You don't …

Why the client-side for grdapi Cannot be on Laptops

But if you had a Jump Server...

Collectors/CM

Jump Server
script_1.cmd
script_1.sh
script_1.log

Laptop

Laptop

Laptop

This Jump Server does not need to be large. A small Linux instance is good enough. Only needs to be able to access the Guardium appliances

Example of Organization I used successfully on jump server

I created a folder for each collector and a folder for the scripts, as follows:

```
/
├── /scripts ──── /archive
├── /coll_1 ───── /archive
├── /coll_2 ───── /archive
└── /coll_… ───── /archive
```
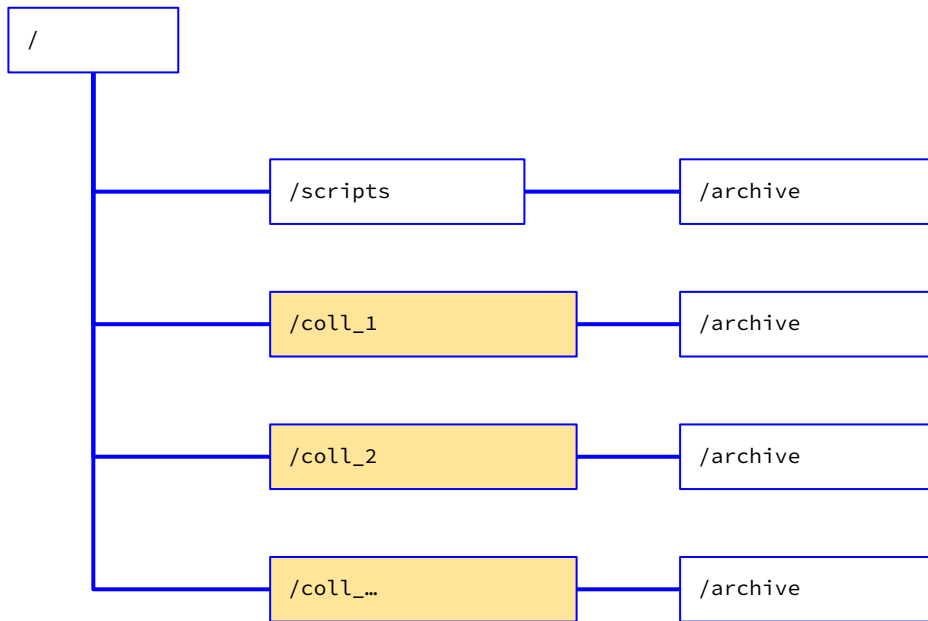
On a Linux machine, you create a shared account to access this directory structure, an account that can be connected to only with sudo.
The entire system becomes securely shared, unique, 24*7 365 available, backed up by the company etc… You can focus on managing the system and automate, automate, automate

In Archive you put older log files

Examples of Scripts and how to execute them

In the common script folder, you can have text files with grdapi you want to execute commonly.
Example : in **show.cmd**

```
show system cpu profile
```

In each folder for collector, you create a simple script **show.sh** with chmod 755, calling those scripts as follows:

ssh cli@collxxx**<../scripts/show.cmd>**show.log

cat **show.log**

```
show → .cmd
     → .sh
     → .log
```

You can then execute the script as **./show.sh** and the output will be displayed to you (the cat) AND recorded in **show.log**

Depending on whether you have installed the authentication with **public key** or not, you will be prompted for the **password** or not

You will **need to rename the show.log** into something else before running the script again or it will be erased and you will lose the previous run …up … ;-)

Example of Organization I used successfully on jump server

I created a folder for each collector and a folder for the scripts, as follows:
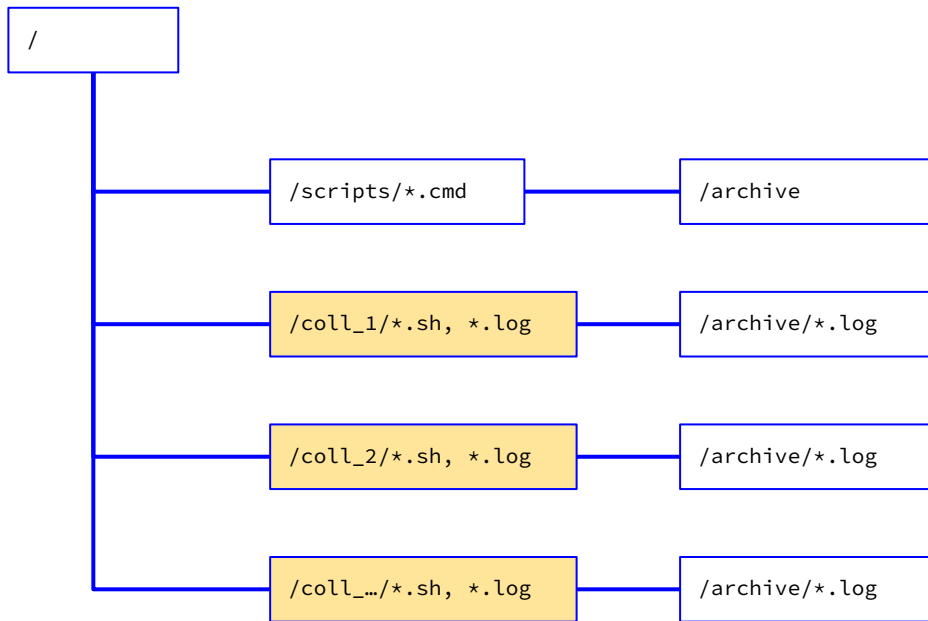
How to contact us

info@context22.com
Support@context22.com
LinkedIn
www.context22.com
(under construction)