Bootstrapping Data Science in a Large Organisation

How FOSS and containers can help us do real work

Riaz Arbi
2019-02-20 (updated: 2019-03-13)

Topics

- Is the organisation prepared for data science workflows?

- Getting work done

- Building a free, scalable, enterprise-grade data science environment

Is the organisation prepared for data science workflows?

Why do companies hire data scientists?

- Data-drive decision making: Increase efficiencies with (regular) statistics
- Automation: Increase productivity by automating parts of workers' jobs
- Exploration: Learn something about their business they didn't know
- Experimentation: "We don't want to get left behind"
- Unify large silos: Get underneath the BI layer, connect different large systems
- To push the technical edge: IoT | agile | cloud | data lakes | Blockchain | AI

Have we got the necessary priors?

- Data-driven decision making: Do we have software that does this math?
- Automation: Do we have triggering systems on place?
- Exploration: Can we connect to our underlying databases?
- Experimentation: Has the business performed an 80/20 exercise?
- Unify large silos: Is there DB admin skill to open these dbs up?
- To push the technical edge: How to owners of exisiting technological stack feel about this?

Getting work done

Getting work done

Traditional Corporate IT concerns -

- Multiple versions of the truth: departments choose the math that flatters them
- Technical debt: data science tools are notoriously fast-moving
- Security: arbitrary code execution!
- Stability: spaghetti code processing pipelines
- Preserving inter-departmental interoperability: proliferation of `.R`, `.py`, `.ipynb`, `.Rmd`, `.html` files.

These are important considerations!

Getting work done

Typical Constraints -

- Machines have to use enterprise software [MS Office / Sharepoint / SAP]
- No admin access to laptops
- If it is on the corporate network, IT totally controls the stack
- Server internet access is strictly controlled

Solution 1: BYOD

Flash drives, and your macbook from home

1. Insecure
2. Compute limited
3. May be against the rules
4. How will you possibly productionise?

Getting work done

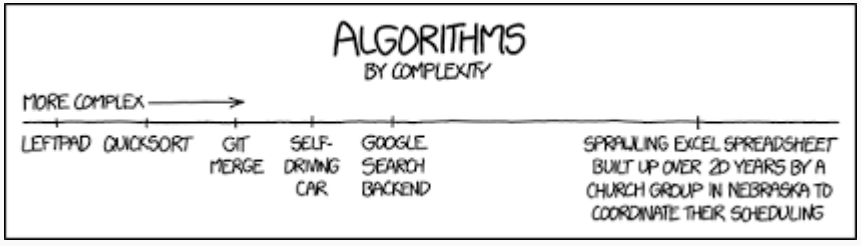
Typical Constraints -

- Machines have to use enterprise software [MS Office / Sharepoint / SAP]
- No admin access to laptops
- If it is on the corporate network, IT totally controls the stack
- Server internet access is strictly controlled

Solution 2:

Wait for the organisation to catch up

1. Work on Excel
2. Manually download data using enterprise application UI
3. Work with IT to refactor models into enterprise-compatible code



We can do better

How do we do *good* data science, that *has impact*, in a *low-cost*, *low-risk* way?

Build a system that does everything you need it to

- Don't distract other departments from their work
- Don't hire anyone else
- Don't compromise existing systems
- Don't use the cloud, don't spend any money or pay for any licenses!

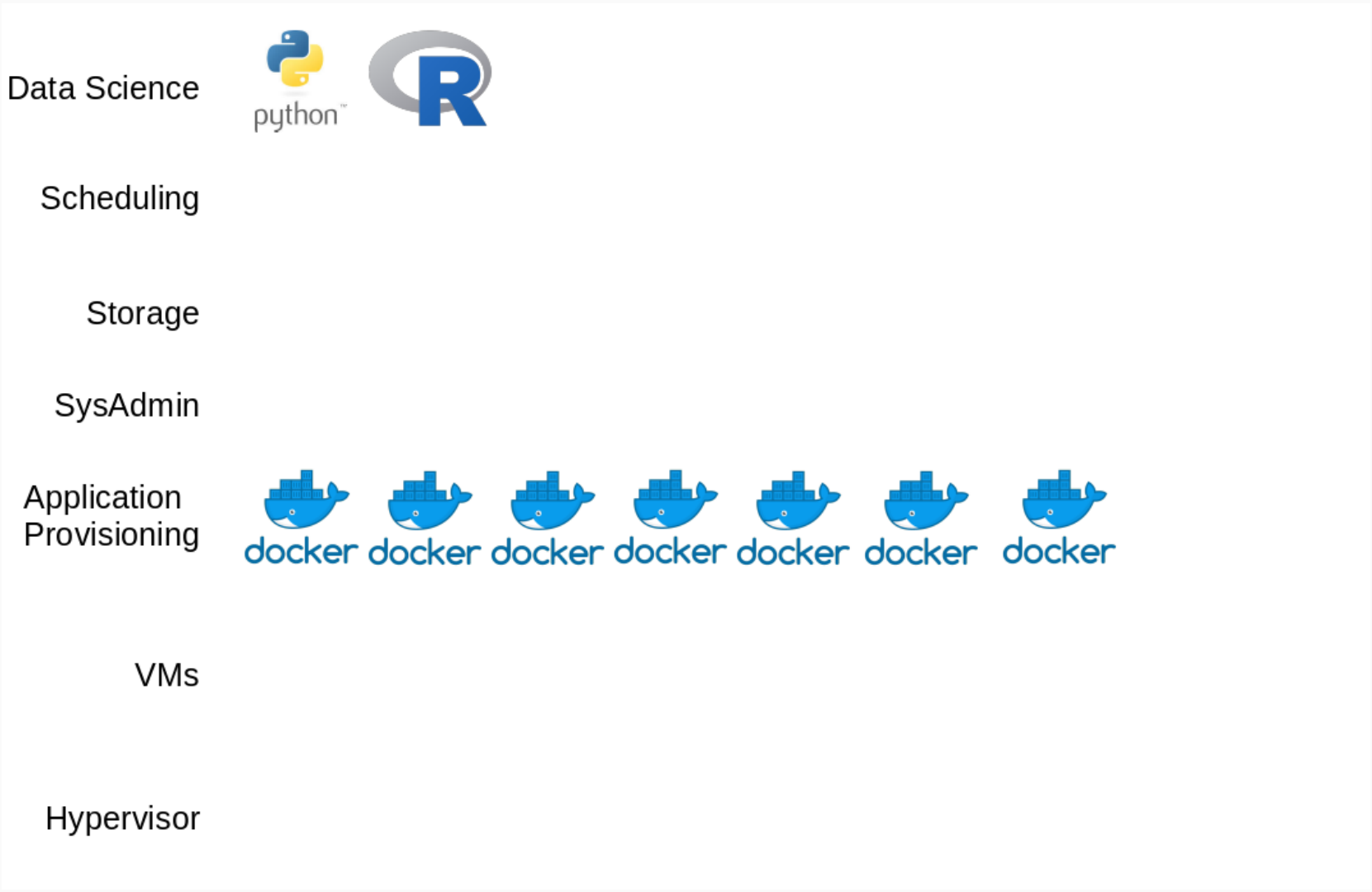
Achievable?

Building a free, scalable, enterprise-grade data science environment

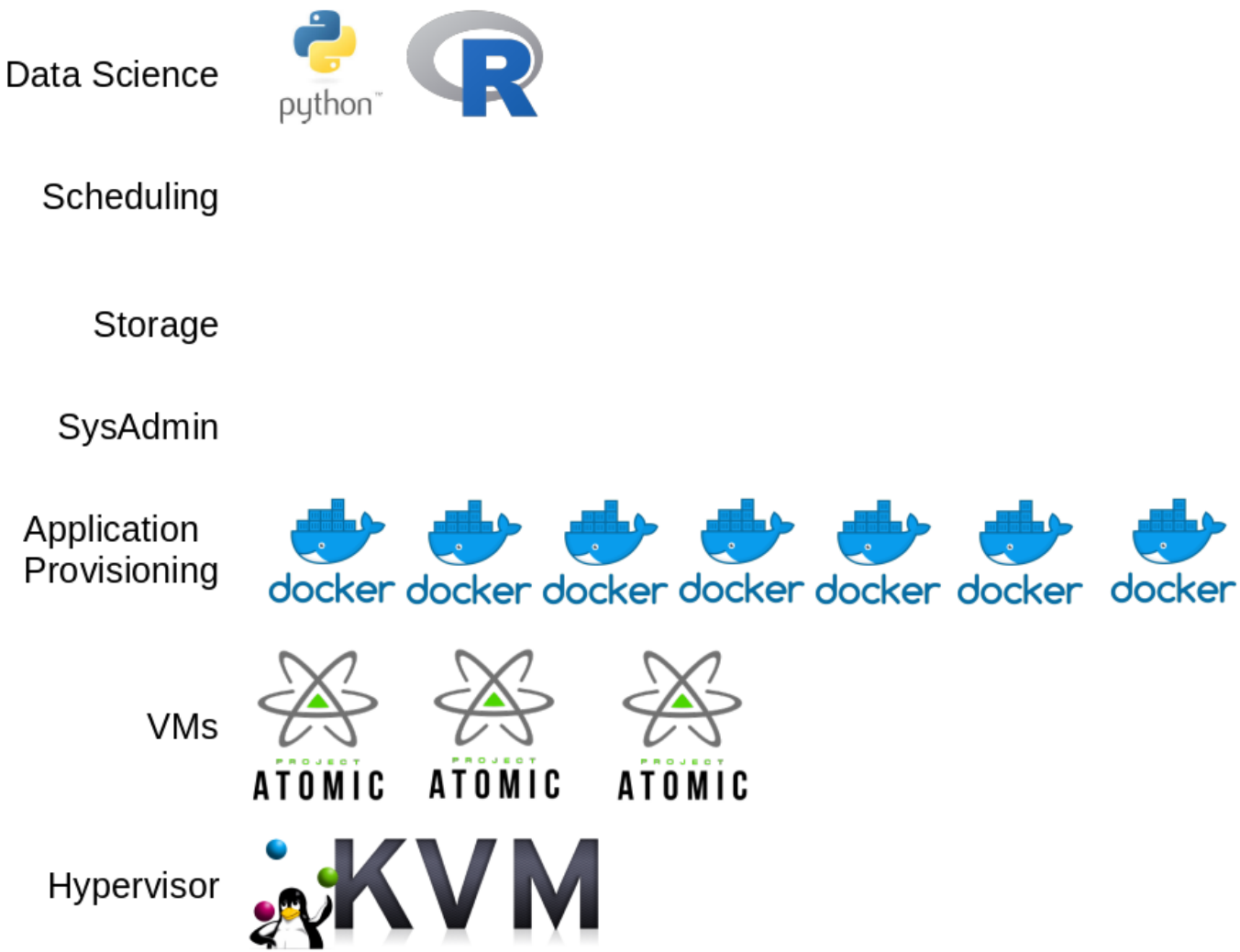
Ideal system requirements

- *Good interoperability:* can our systems talk to each other? Can they talk to anybody?
- *Large, unstructured storage with fine-grained control:* can we dump data somewhere? can we secure access? can we make it available to anybody on the network?
- *Code repositories:* can new work be versioned? Multiple collaborators? Branching for experimentation?
- *Code review:* do we have regular over-the-shoulder conversations about what our developers are doing?
- *General purpose job scheduling:* do we know how to fire scripts
- *Disposable, scalable environments:* how long does it take to give someone an isolated environment with 2 cores and 2GB ram? What about 16 cores? What about 64?

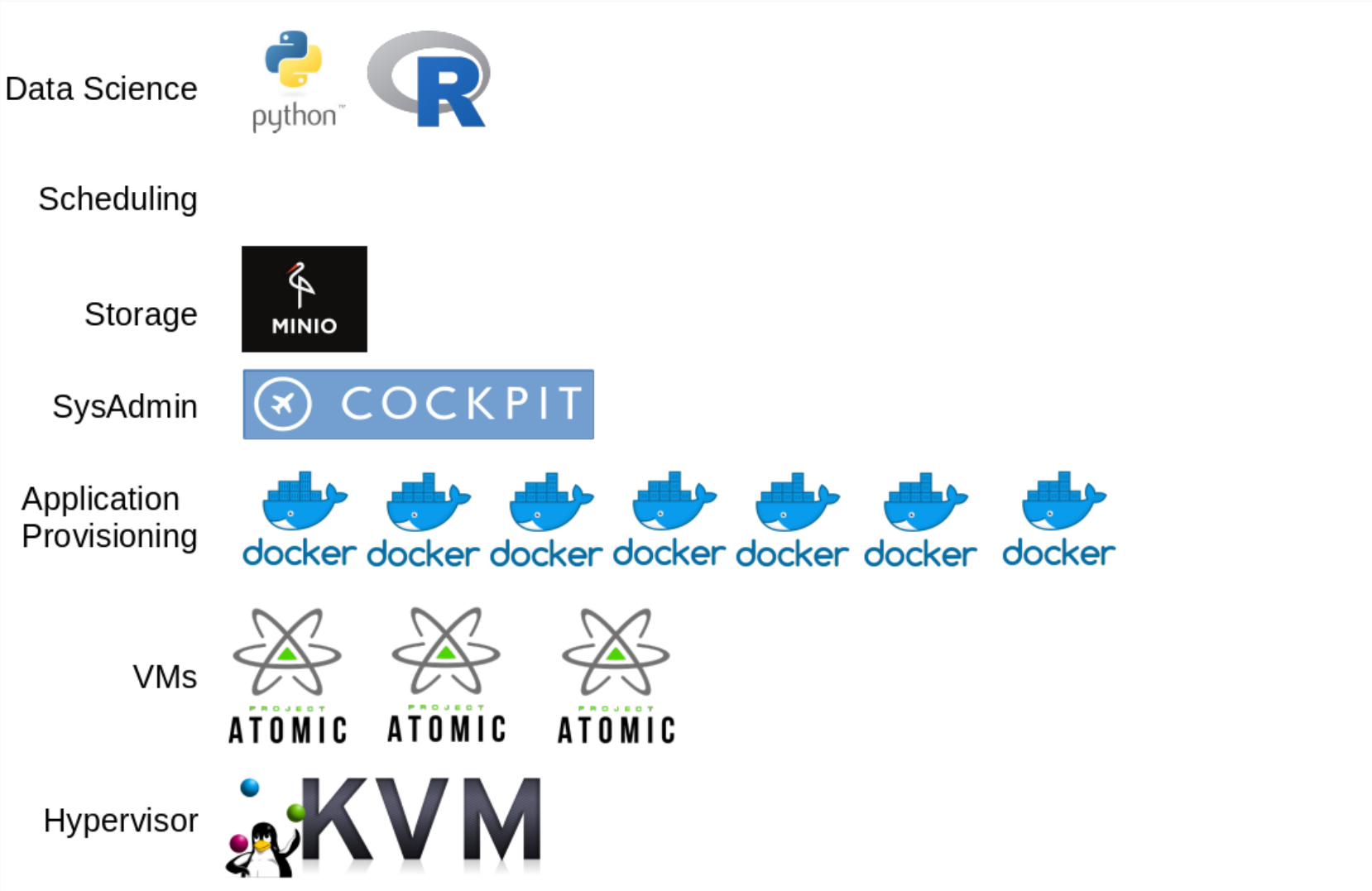
Roadmap: Get started on a laptop



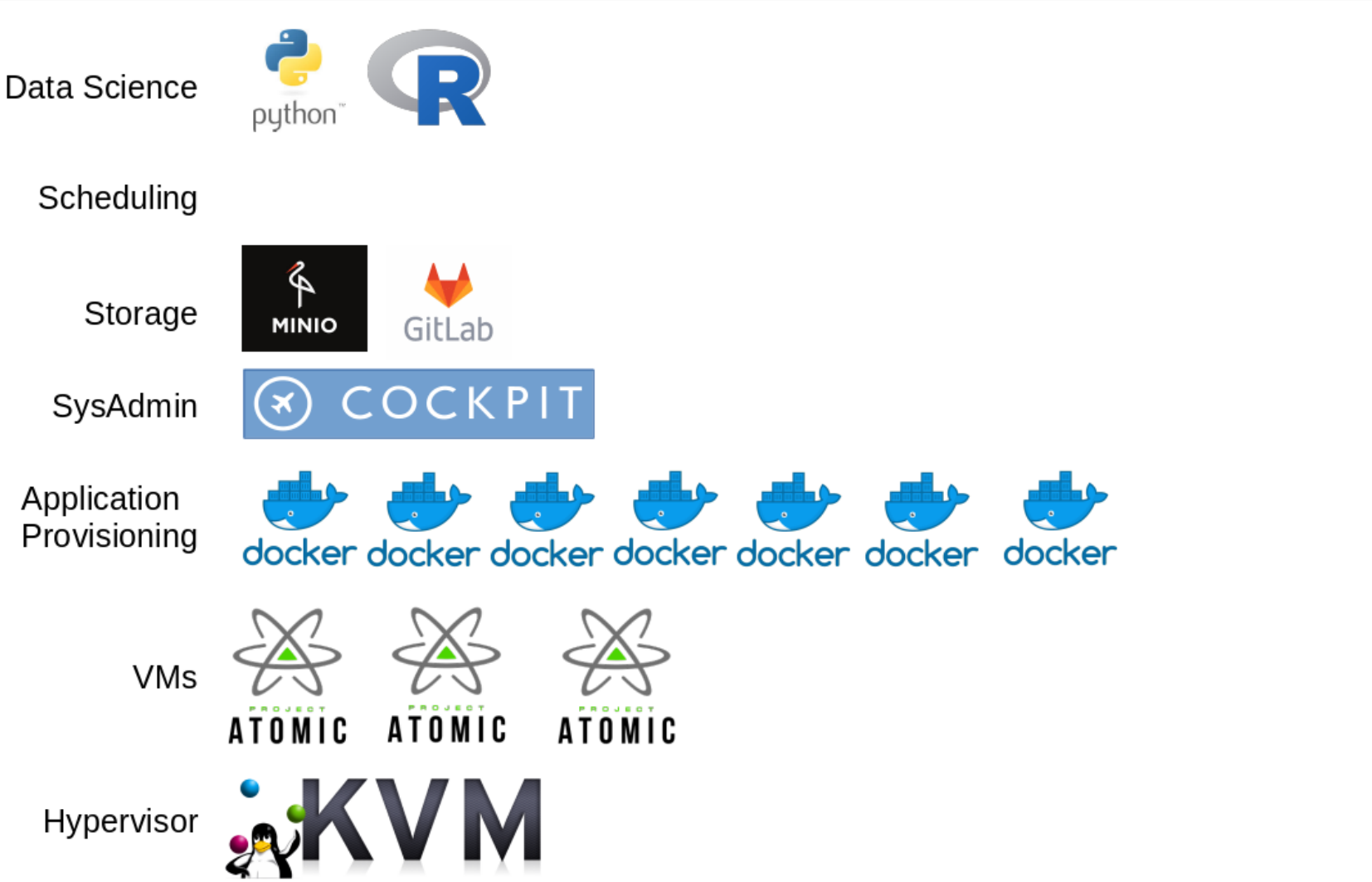
Roadmap: Get an isolated server



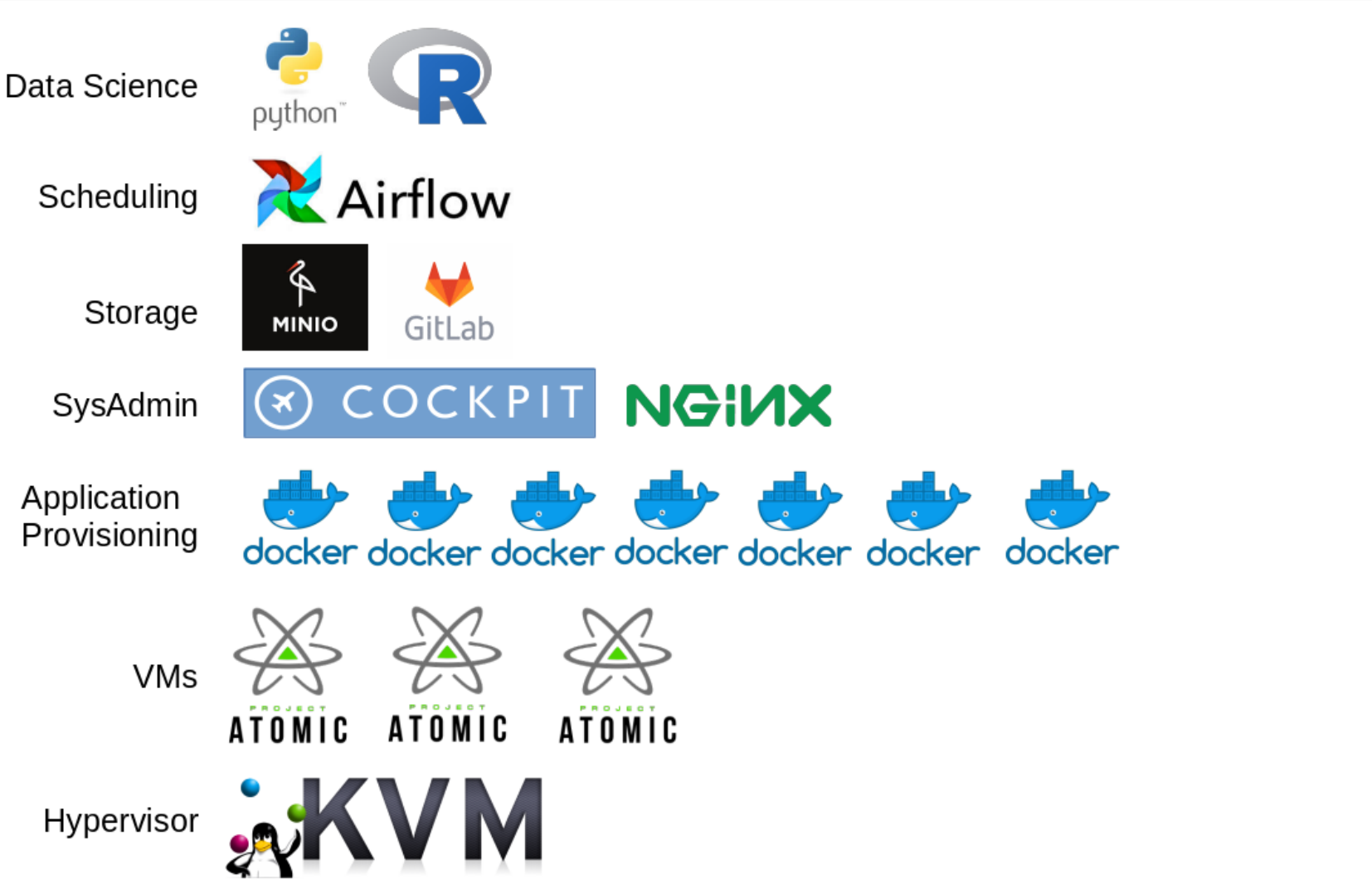
Roadmap: Add storage capacity



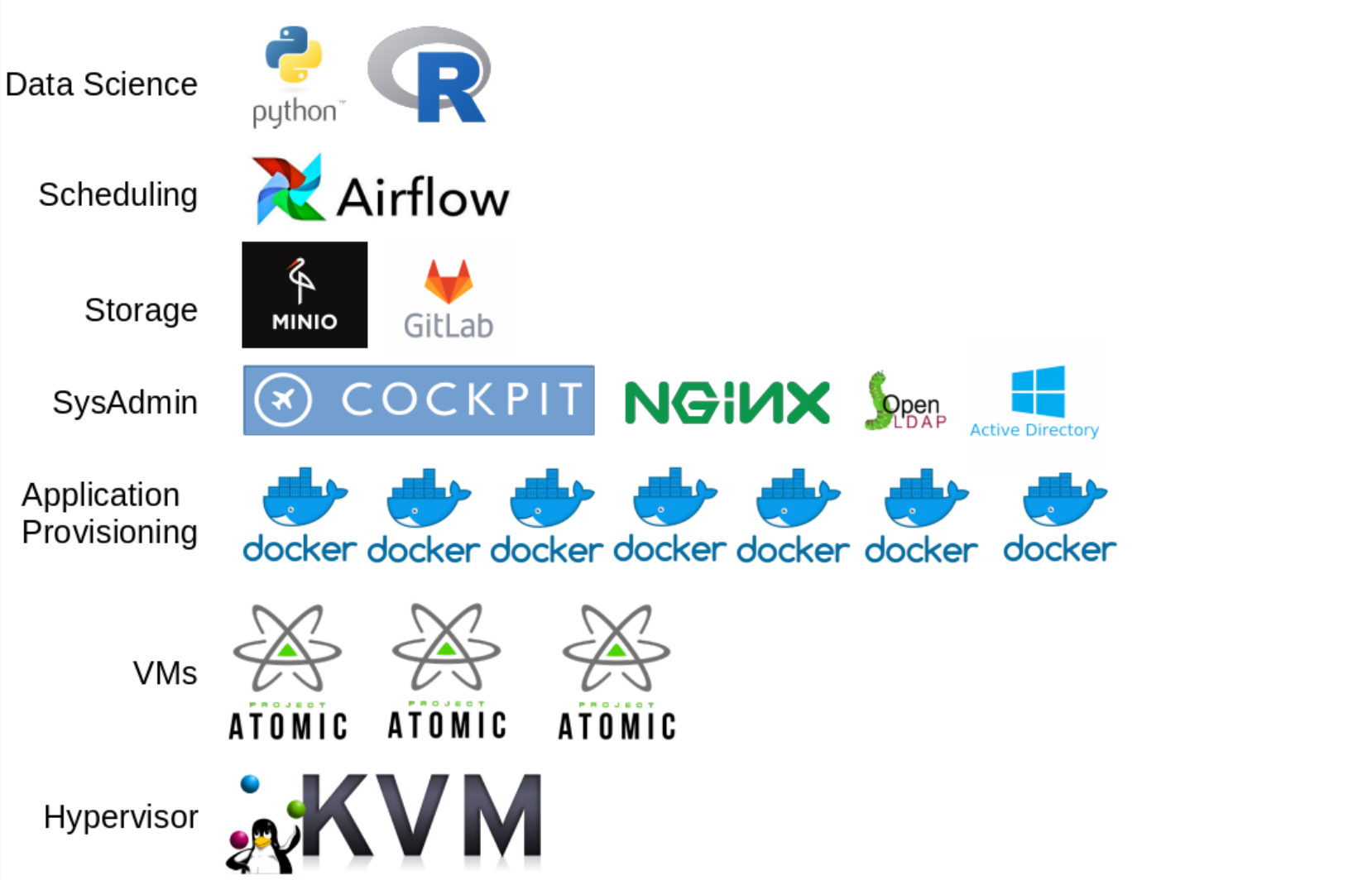
Roadmap: Add good code practices



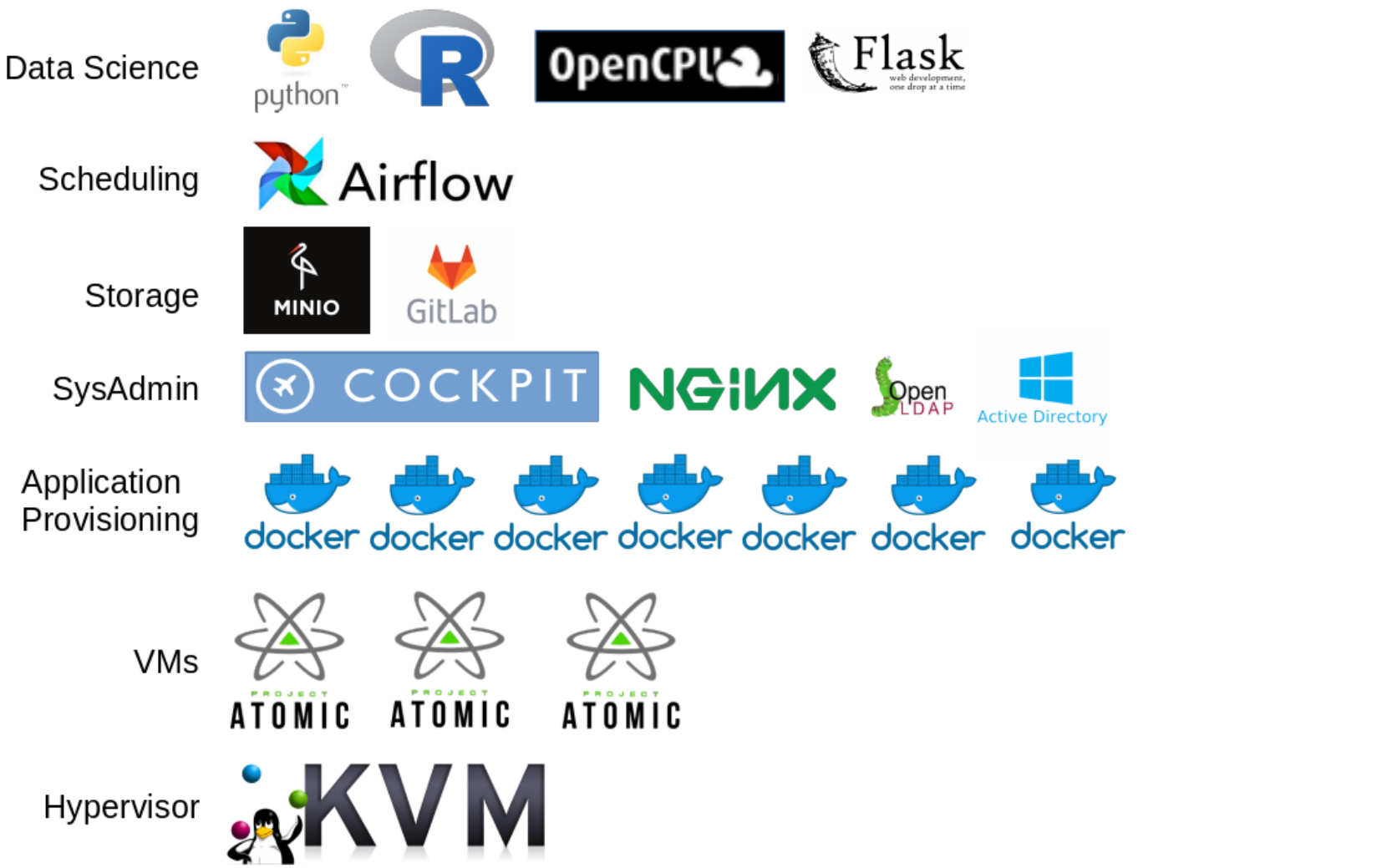
Roadmap: Start building pipelines



Roadmap: Unify auth



Roadmap: Integrate



Back to IT Concerns

- **Multiple versions of the truth** -> Pipelines + code version control = auditability
- **Technical debt** -> Provide access to model predictions via API; swap code underneath if needed. All tools are fully cloud compatible and easy to migrate.
- **Security** -> Isolated from corporate network. Isolated into VMs. Isolated into containers. Code transparently housed in git.
- **Stability** -> Isolation and limiting of processes via container.
- **Preserving inter-departmental interoperability** - All UI access via browser (not IE though, Jupyter!)

Costs

Money

- *Hardware:* R0 for decommissioned server or cluster of decommissioned consumer towers; Up to ~R1m for a fully loaded server.
- *Software:* R0
- *New hires:* R0

Time

- *Admin load:* A day a week

Obligations

- This is Open Source! Participate. Ask questions, help with documentation, submit feature requests.

Enterprise support

If you *reallly* want to spend money

- KVM -> Get enterprise Linux (Red Hat / SUSE / Ubuntu etc)
- Docker CE -> Docker EE
- Gitlab EE -> Activate License
- Minio -> Minio SUBNET
- RStudio Server -> RStudio Server Pro
- NGINX -> NGINX Plus

...and all the consultants you can afford

Now we can get to work!