

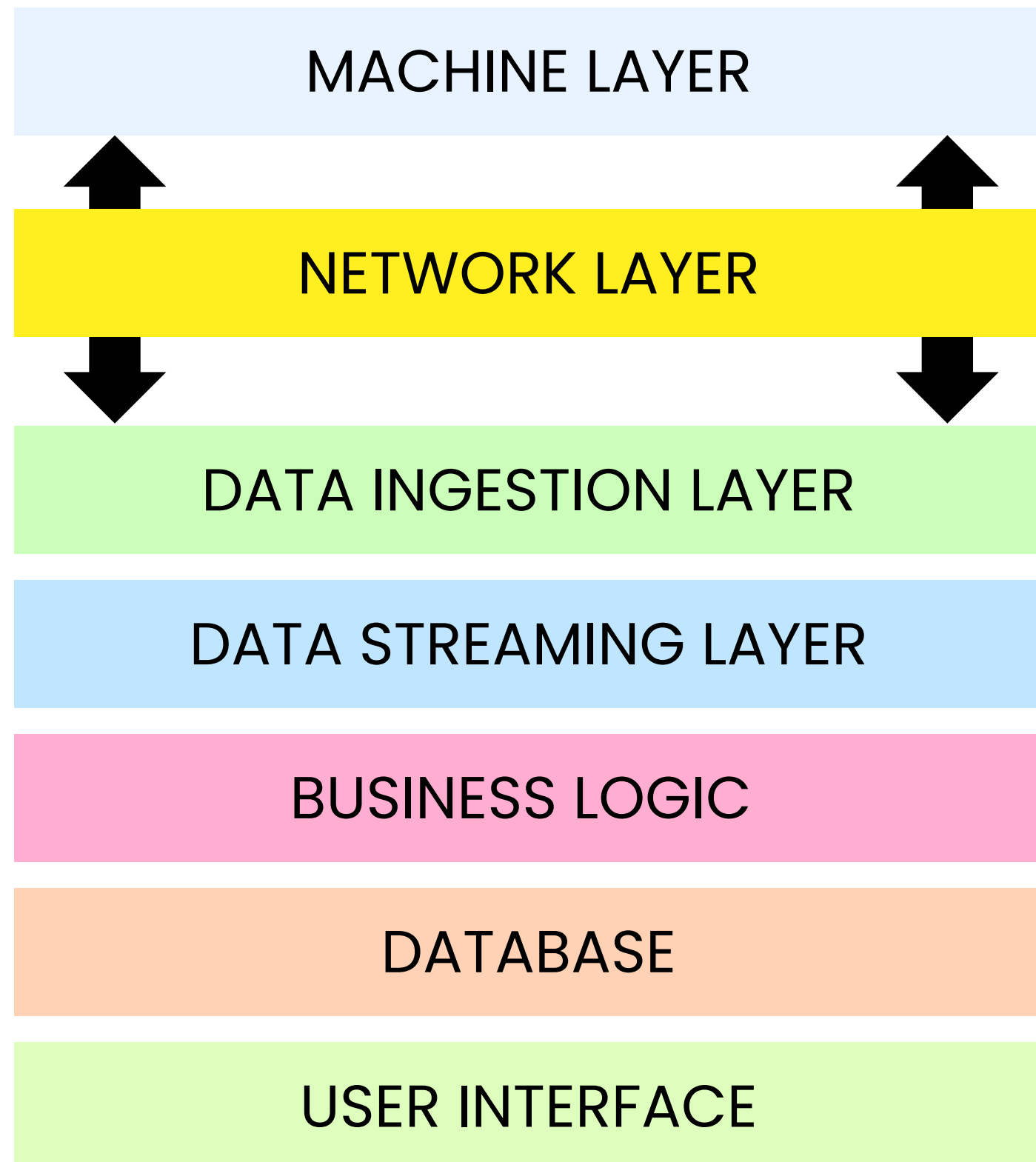
MOTOLITE[®]

END-OF-OJT REPORT

NEO ELROND V. CABRERA, 07/28/2025


SCOPE OF INTERNSHIP

Industry 4.0 monitoring platform



MAIN TASK: Register and **set up** machines for the data ingestion layer of the platform.

Using the network, the data ingestion layer **automatically and directly** extracts data from the machines. **No ingestion, no automatic data.**

	Machine count
Year-end target	137
June 19 (start of internship)	57
July 25 (end of internship)	78 (21 )

MAIN GOAL

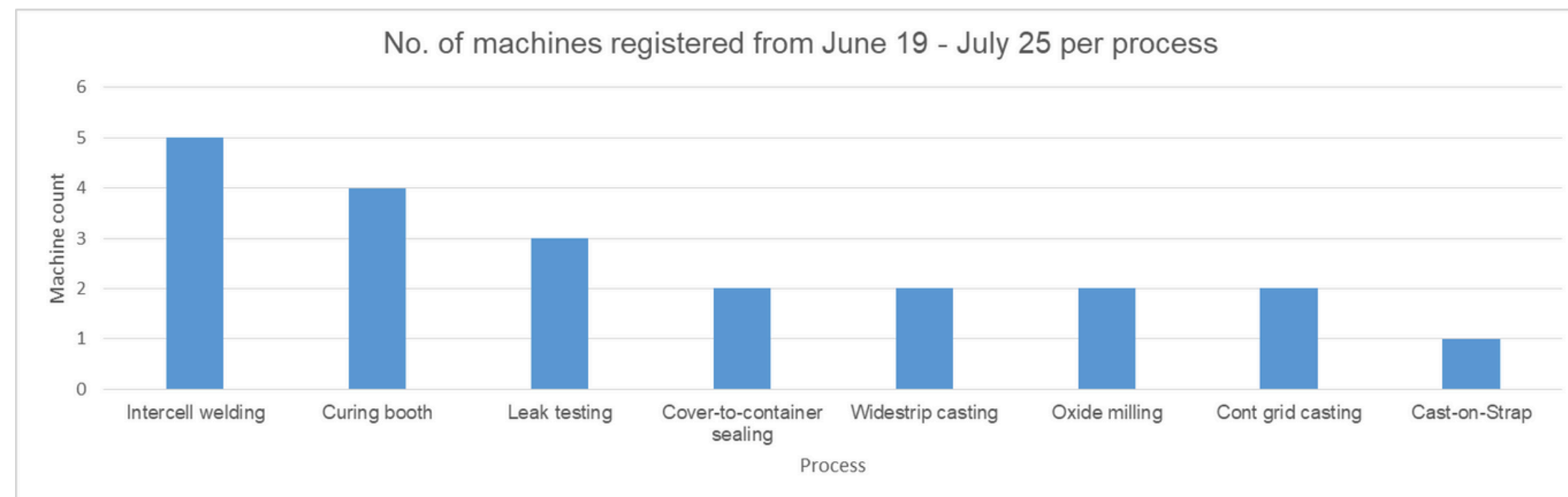
registering machines into the platform

Breakdown by process

Process	Machine count		
	June 19, 2025	July 25, 2025	Difference
Intercell welding	5	10	5
Curing booth	17	21	4
Leak testing	1	4	3
Cover-to-container sealing	1	3	2
Widestrip casting	4	6	2
Oxide milling	7	9	2
Cont grid casting	0	2	2
Cast-on-Strap	9	10	1
	44	65	21

← 4 ICW PLCs are Mitsubishi PLCs, which were not yet supported until July 23. **I helped develop / test code for it.**

← **47.73%** increase



PROBLEMS ENCOUNTERED

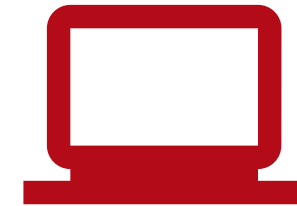
data ingestion layer



1. Machine registration is **error-prone**, taking up **10-15 minutes**



2. Data ingestors are **non-modular** / **hard to update**



3. Data ingestors are **non-portable**



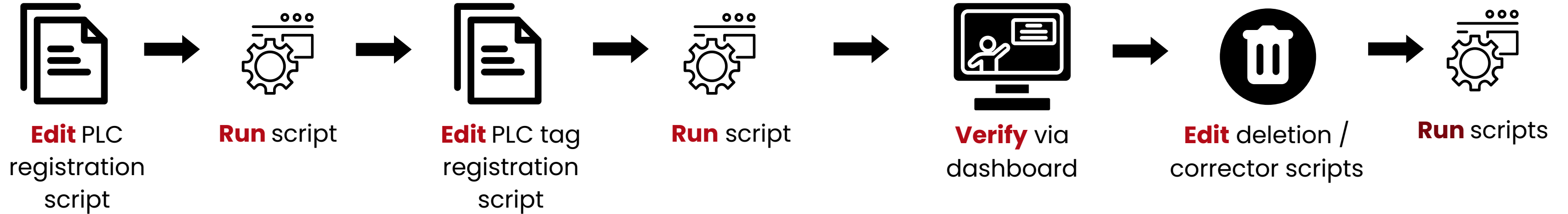
4. Data ingestors frequently disconnect / crash, lasting only for **3-7 hours**

PROBLEM #1

error-prone machine registration

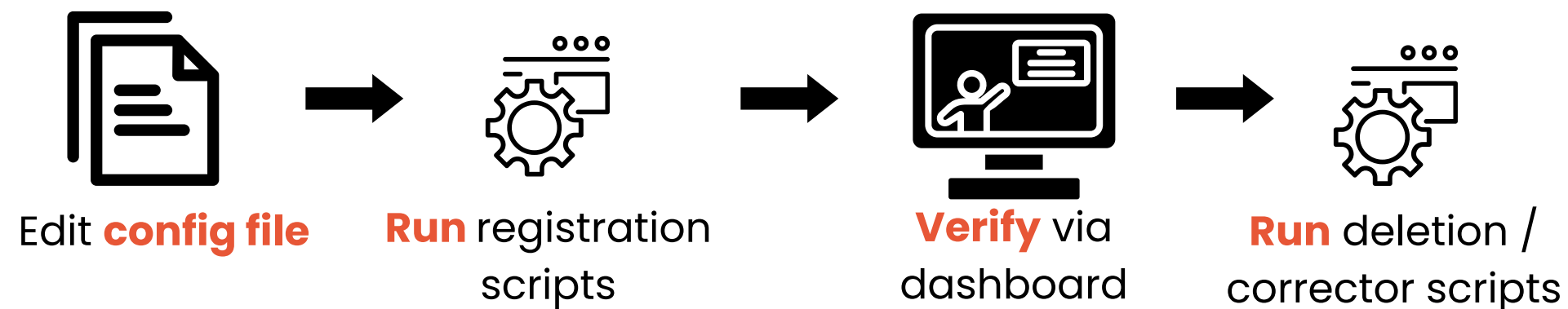
PROBLEM It is **tedious, redundant**, and takes approx. **10-15 minutes**

OLD PROCESS



SOLUTION Simplify the process by using a **single config file** to update all scripts

NEW PROCESS



IMPACT

less **human error** when editing,
only uses **approx. 5 minutes**

PROBLEM #1

error-prone machine registration

IMPACT

722 machines are still unregistered. How much money can be saved with the new registration process?

	Process	
	Old	New
Time per machine (mins)	10	5
Total time (hours)	120.33	60.17
Est. manpower cost (₱/hr)*	300	300
Cost (₱)	36,099	18,051

**Based on current salary rates for entry-level data science engineers*

50% decrease in costs (w/ zero human error)

PROBLEM #2

non-modular ingestors

PROBLEM

the old style of coding new ingestors is unscalable and redundant

previous coding style

copy-paste existing code from another machine (ex. ICW)



edit a small part to adapt it to new machine



leave the rest of the code intact

To update all scripts, we have to **edit them one by one**

SOLUTION

modular coding style

current / improved coding style

code a **"module"** or **"guidebook"** that says what an ingestor is in general



use module for new machine code



focus on writing code unique to new machine

If the module changes, **all ingestors** change at once

IMPACT

Ingestion code is now **easier to maintain / update**

PROBLEM #3

non-portable data ingestors

PROBLEM

ingestors only work on one computer; migration to another PC needs **a full reinstall**

SOLUTION Containerization

Place code and dependencies in a **container** – a box that can run anywhere as long as a specific software application (industry standard: **Docker**) understands it.

ANALOGY: Semicon companies do not ask their customers to assemble an IC or a computer chip. The chip or IC is ready to use – **containerized**.

PROBLEM #3

non-portable data ingestors

IMPACT

In 130 days, PBI will migrate its monitoring system to the cloud to have better resource / connectivity management. Why containerize it first?

	No containerization	W/ containerization
Installation	Reinstall all code manually	Just install Docker, click run on container
Cloud support	Unguaranteed	Industry standards like AWS or Azure integrate it
Response time	depends on the platform	Half of non-containerized apps (IBM, 2021)
Cloud server costs	heavyweight → more expensive	75% cost reduction (IBM, 2021)

Containerization (and Docker) is forward thinking.
It reduces **future tech debt**.

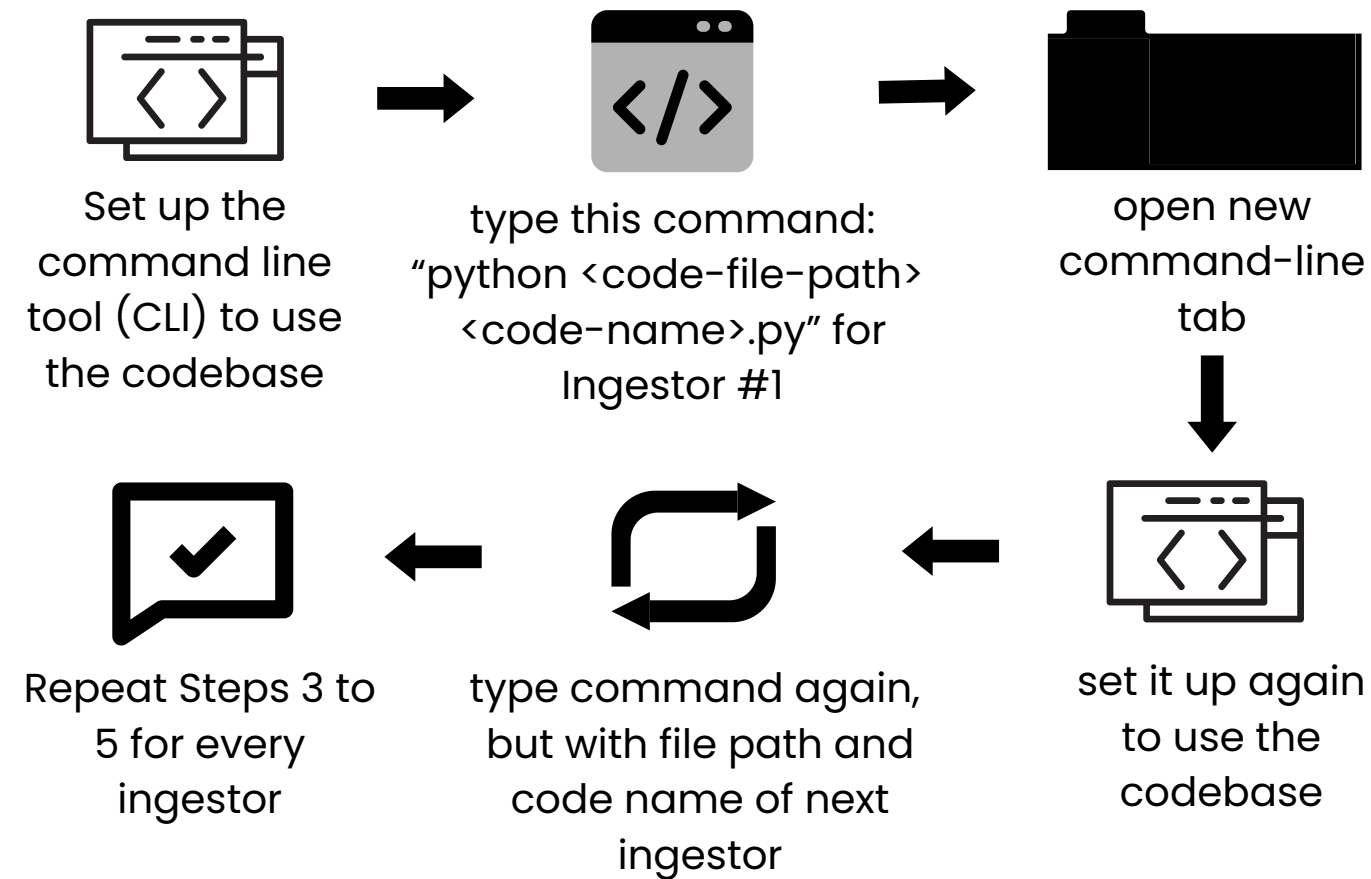
PROBLEM #3

non-portable data ingestors

IMPACT

How to run ingestors for each process with and without containerization?


W/O CONTAINERIZATION




```

(apc-rules-py3.10) PS D:\apc-rules> python .\ingestors\cast_on_strap\cos-ingestion.py
2025-07-24 11:40:11.538 | INFO | apc_rules.infrastructure.db.sql_server:get_engine:24 - Connected to SQL Server at m
sssql+pyodbc://sa:pmdata@science@172.31.3.40:1433/APC?driver=ODBC+Driver+17+for+SQL+Server&MARS_Connection=yes
2025-07-24 11:40:13.784 | INFO | __main__:<module>:164 - Ingesting data from ABA Line 02 COS...
2025-07-24 11:40:13.793 | INFO | __main__:<module>:164 - Ingesting data from ABA Line 03 COS...
2025-07-24 11:40:13.796 | INFO | __main__:<module>:164 - Ingesting data from ABA Line 04 COS...
2025-07-24 11:40:13.800 | INFO | __main__:<module>:164 - Ingesting data from ABA Line 05 COS...
2025-07-24 11:40:13.807 | INFO | __main__:<module>:164 - Ingesting data from ABA Line 06 COS...
2025-07-24 11:40:13.812 | INFO | __main__:<module>:164 - Ingesting data from COS MACHINE,TBS ,LINE 12...
  
```

W/ CONTAINERIZATION


Open Docker


Click play button
on the "ingestors"
container

	Name ↓	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	ingestors	-	-	-	3.76%	4 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	widestrip_ingestor-1	2d3aaeee6dc9	ingestor-base-image		0.74%	4 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	plate_formation_ingestor	84b96f731699	ingestor-base-image		0.07%	4 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	oxide_milling_ingestor	69f63157f7e2	ingestor-base-image		0.07%	4 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

PROBLEM #4

machine-server connectivity issues

PROBLEM

Connectivity issues either crash or corrupt ingestors. There is a high possibility of the ingestors crashing and disconnecting, producing no data until the next manual restart.

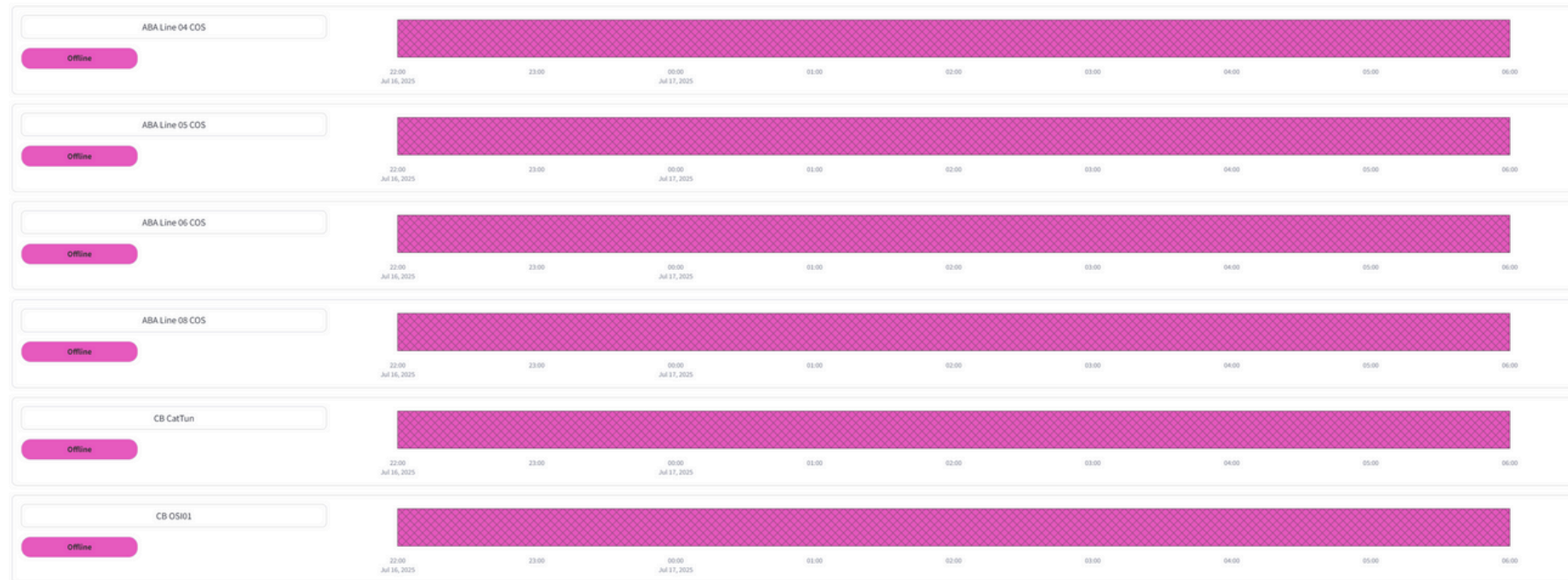
24/7 OPERATION NOT GUARANTEED

Causes	Solution
Connectivity errors	1. Refresh / relaunch ingestor after 5 failed connections 2. Long-term: Coordinate with KAISA / IT Department
No self-reboot for ingestors	Reboot mechanism
Database errors	Rollback mechanism
Network resource exhaustion	1. Temporary: Reboot ingestors every hour 2. Long-term: migrate to the cloud for better network resources and management

PROBLEM #4

machine-server connectivity issues

BEFORE (JULY 16, 10 PM – 6 AM)



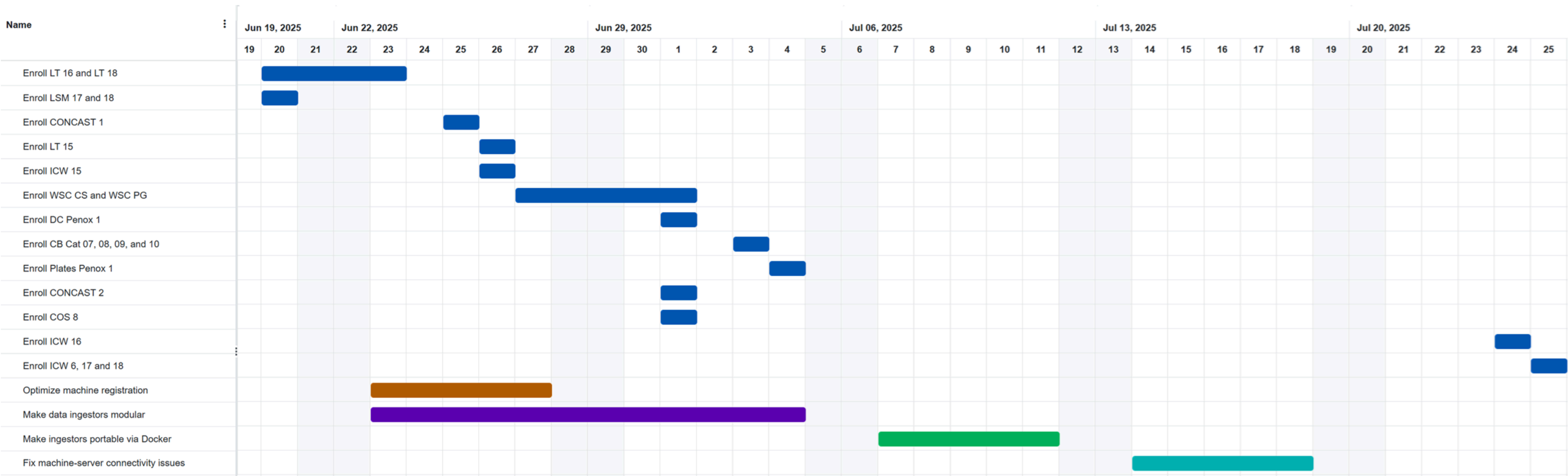
AFTER (JULY 17, 10 PM – 6 AM)



Ingestors can now more reliably produce data for **24 hours**, even without personnel monitoring them.

TIMELINE

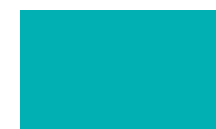
summary of tasks



main task



make ingestors
modular



fix connectivity
issues



optimize machine
registration

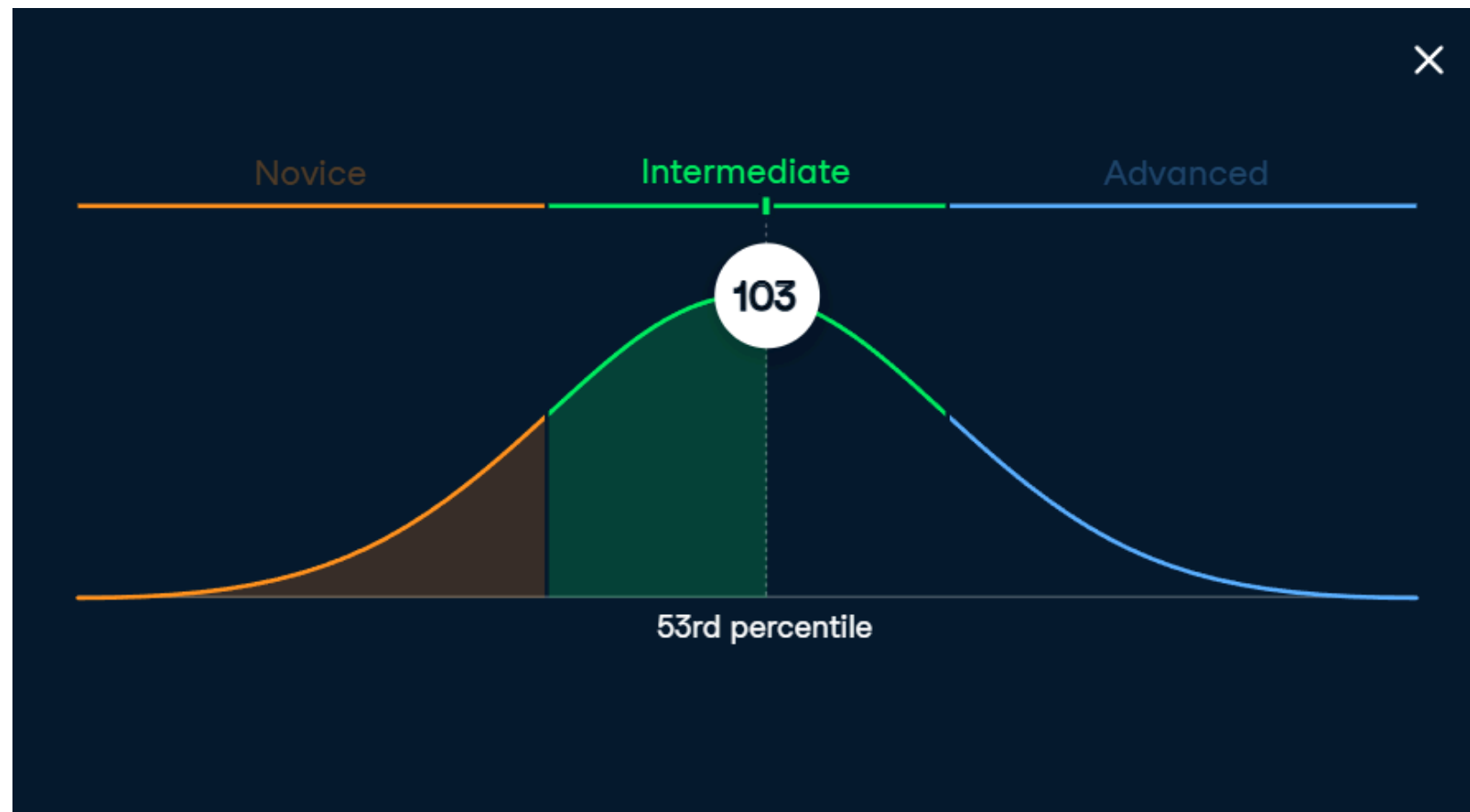


make ingestors
portable

PERSONAL IMPROVEMENT

proficiency with the Python language, coding

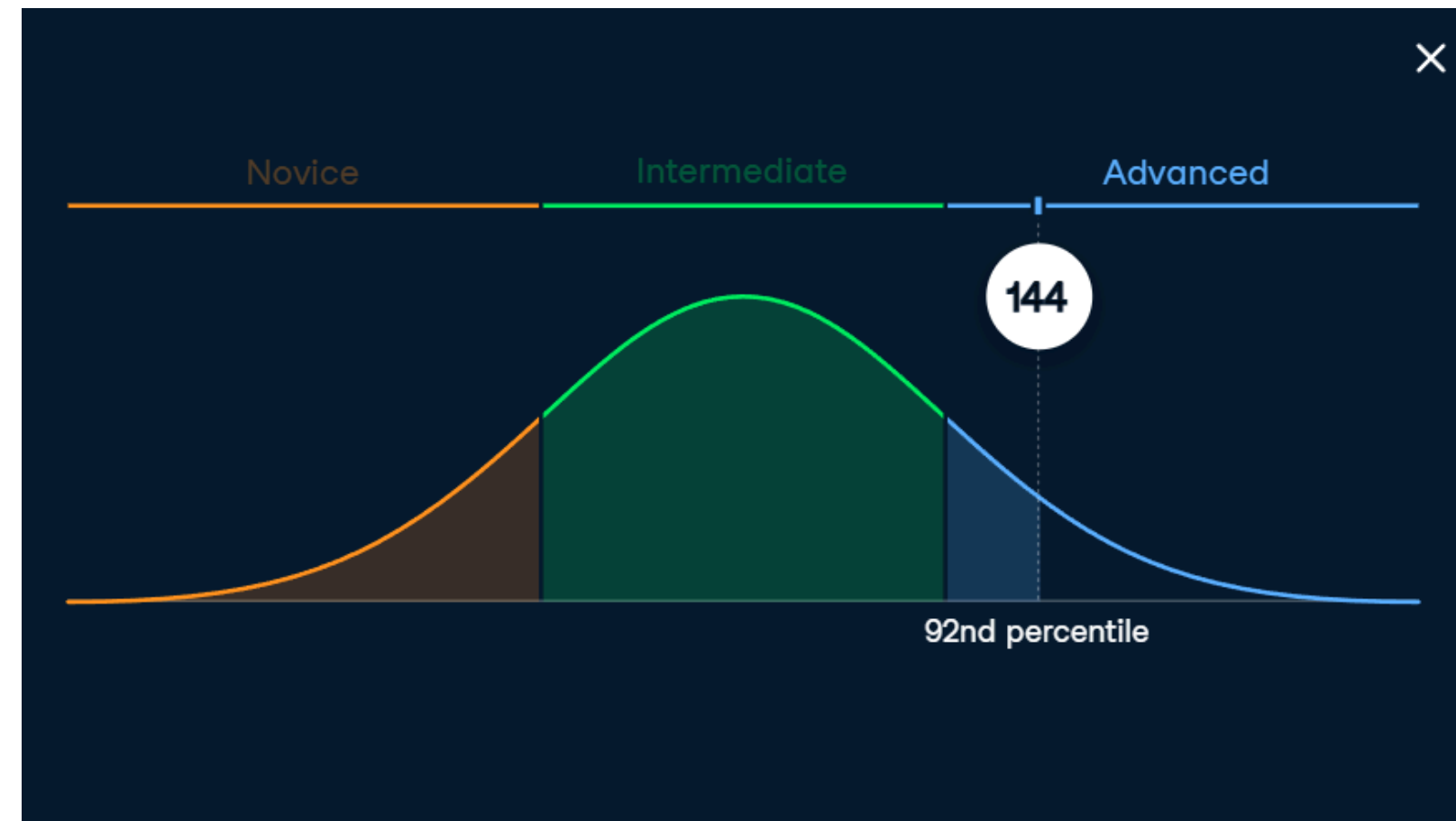
BEFORE (JUNE 25)



You received a score of 103. You performed better than 53% of all others that have taken this assessment.

[Review Your Answers](#)

AFTER (JULY 16)



You received a score of 144. You performed better than 92% of all others that have taken this assessment.

[Review Your Answers](#)

Important lessons: writing clean, maintainable industry code via Python

PERSONAL IMPROVEMENT

BEFORE (JUNE 19)

1. Too technical and wordy with reports
2. Inexperienced with industry standards and tech
3. Not used to inter-team communication within a factory context
4. Small engineering network

AFTER (JULY 25)

1. More guided and careful with reports / presentations
2. Now knows PLC communication, Docker, SQL, Python, Apache Kafka
3. Better communication skills
4. Larger engineering network

REFERENCES

IBM (2021). *Virtual Machines versus containers How IBM WebSphere Hybrid Edition with Red Hat OpenShift can lower server costs and improve response time*. Retrieved from <https://www.ibm.com/downloads/documents/us-en/107a02e95c48f737>