Stop e K8
like a er.

TALK.CONTENTS.TXT

← → ↻ http://localhost:3000/kubecon-2026-NA

NOT Kubernetes 101.

“Communication about infrastructure fails when we explain implementation details before establishing mental models and relevance.”

Loading …







KubeCon | CloudNativeCon

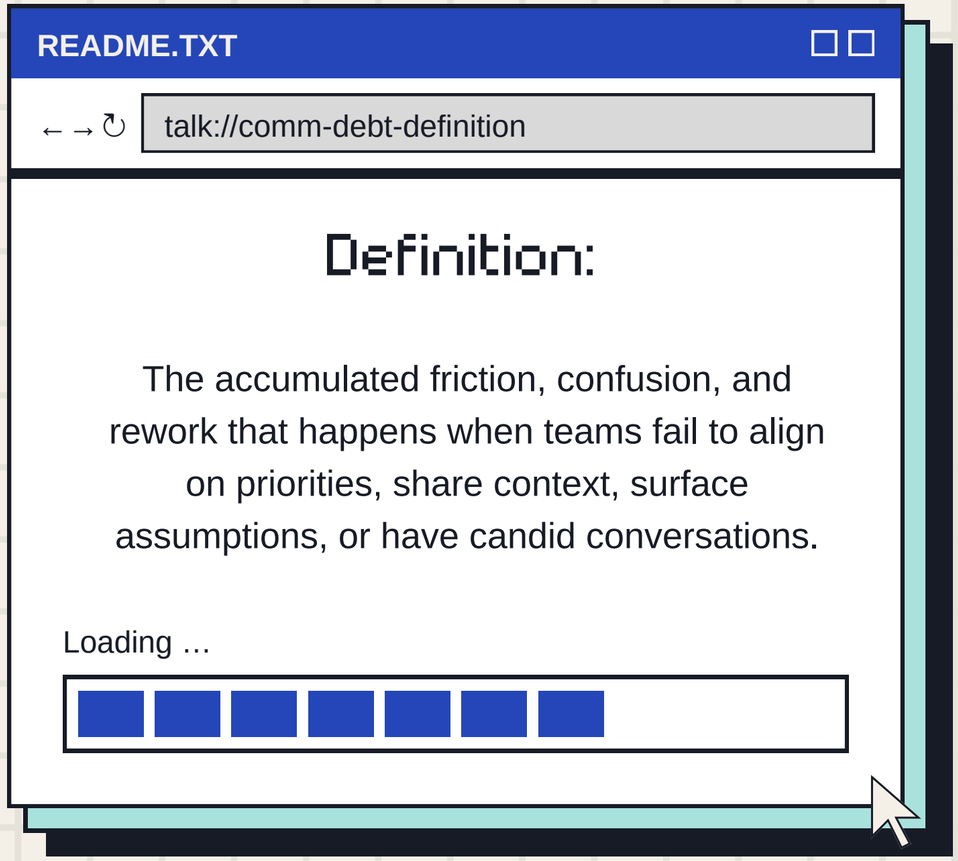
North America 2026

Plain version: ~~~



Communication debt.

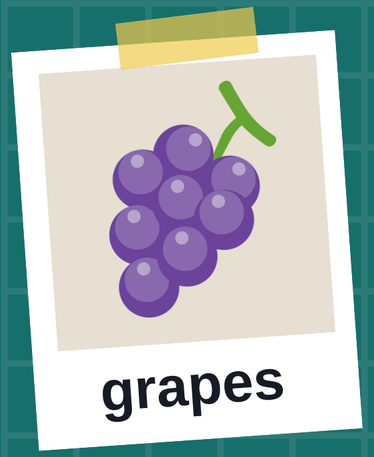
Just as costly as tech debt, but not as visible or tangibile.



Communication debt

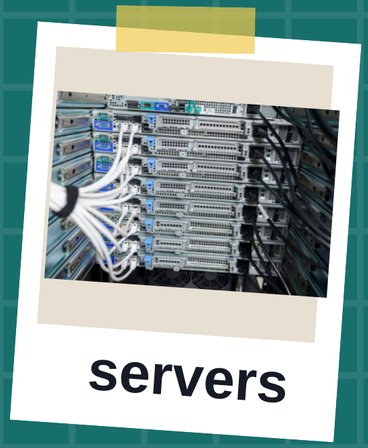
	TECHNICAL DEBT	COMMUNICATION DEBT
HOW IT STARTS	“We’ll clean it up later”	**A nod instead of a question**
WHERE IT SHOWS UP	Bugs, slow builds, outages	**Misaligned priorities, failed adoption, frustrated teams**
HOW IT’S TRACKED	A Jira ticket	**It isn’t**
HOW YOU PAY IT DOWN	Refactor	**Ask one question before you explain**

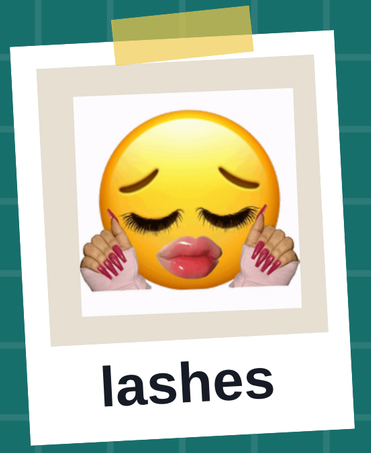
What do these have in common?













They’re all types of clusters...

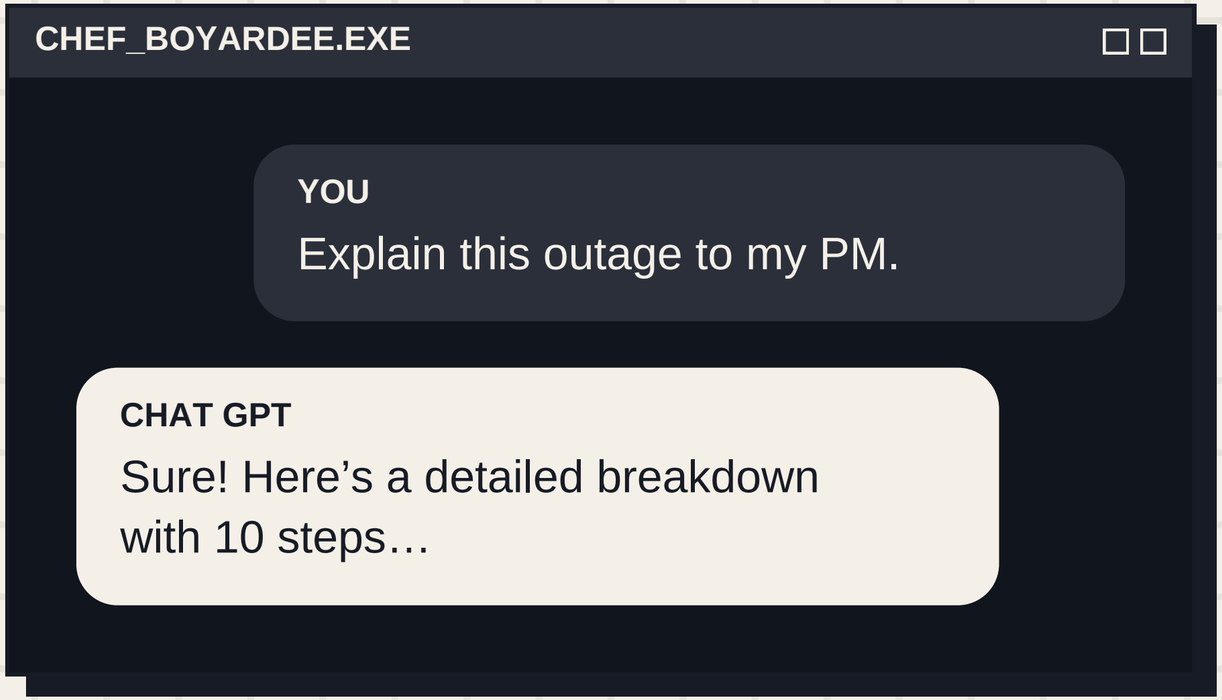
FATAL ERROR

Engineers are inherently cursed.

CURSE.OF.KNWLDG

CURSE_OF_KNOWLEDGE: once you know something, you can’t remember not knowing it.

PRESS ANY KEY TO CONTINUE


CHEF_BOYARDEE.EXE
YOU
Explain this outage to my PM.
CHAT GPT
Sure! Here’s a detailed breakdown with 10 steps…

BREAK THE CURSE

1.JARGON

Jargon should **never** replace an explanation

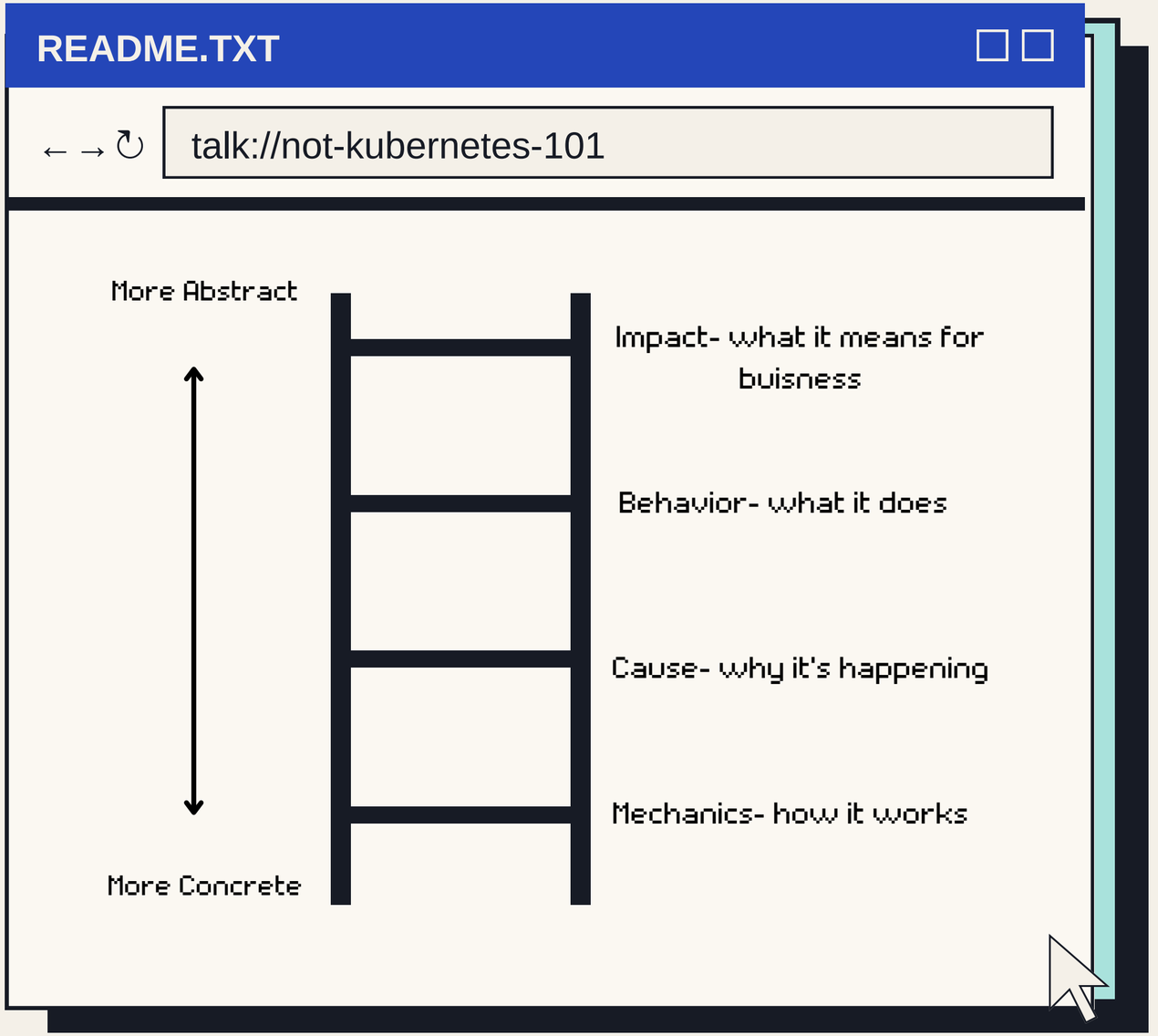
2.QUESTIONS

Ask questions to get context about where they are.

3.THE LADDER

Abstraction ladder coined by S.I Hayakawa in 1939

Abstraction Ladder



Same problem.
Five answers.



EXAMPLE 1

“Today’s release caused an outage. We’ve already reverted, so users are back to normal. The features from that release are offline until we fix what broke. I’ll give you a timeline once we know more.”

PM

THE MOVE

Lead with impact and status.

EXAMPLE 2

“New version broke something on startup, so we’re reverting to the last working version. Think of it like a git revert. Root cause looks like a library mismatch between what we built against and what’s in the container.”

Non-K8s engineer

THE MOVE

Borrow their mental model.

EXAMPLE 3

“It’s crashing on startup, so we’re rolling back. Kubernetes keeps previous versions, so it’s one command. Watch what happens to the pod states when we run it. Logs once we’re stable.”

New engineer

THE MOVE

Keep the jargon, but label it.

EXAMPLE 4

“We pushed an update that broke the app. We switched back to the version we know works. App is stable now.”

Non-technical teammate

THE MOVE

Reassure first.

RUNG 4

Same problem. Four people. Four answers.

4 Teammate “It broke. We switched back. App is stable.”

3 PM “Users are back to normal. Timeline soon.”

2 Non-K8s engineer “Think of it like a git revert.”

1 New engineer · 1→2 “Watch the pod states when we roll back.”

Have you asked a question today?

Ask Question

