PARTICIPANT WORKBOOK

Incident Response Foundations

A practical two-day workshop on detecting, coordinating and learning from production incidents — without blame and without heroics.

Duration · 2 days · 12 hours

Level · Intermediate

Facilitator · Nadia Petrov

Cohort · Autumn 2026 · Cohort 14

PARTICIPANT NAME _____

TEAM _____

DATE _____

BY THE END OF THIS WORKBOOK YOU WILL BE ABLE TO

- Run the first fifteen minutes of an incident with a clear command structure.
- Write a status update that a non-technical stakeholder can act on.
- Separate the timeline of an incident from the analysis of it.
- Facilitate a review that produces changes rather than apologies.

Lesson 1 · The first fifteen minutes

90 minutes

LESSON OBJECTIVES

- Recognise the three decisions that must be made before anything is fixed.
- Assign incident command, communications and operations roles out loud.
- State a working hypothesis without committing to it.

Why the opening minutes decide the outcome

Most incidents are not made worse by the original fault. They are made worse by the twenty minutes after it, when six capable engineers each begin investigating a different theory, nobody is watching the customer impact, and the first status update goes out ninety minutes late because everyone assumed somebody else was writing it.

The opening of an incident is a coordination problem wearing a technical disguise. The technical work is real, but it is not the constraint. The constraint is that a group of people who did not plan to be working together must, within a few minutes, agree on who decides, who investigates, and who talks to the outside world.

This lesson gives you a fixed opening sequence to run every time — the same words, in the same order, whether the incident turns out to be a certificate expiry or a multi-region outage. Fixed sequences feel excessive on small incidents. That is the point: you practise them on small incidents so they are automatic on large ones.

The three opening decisions

Who is in command? One person, named out loud, who does not debug. Their job is to hold the shape of the response: who is doing what, what has been ruled out, and when the next update goes out. If the person in command starts typing into a terminal, the response has lost its coordinator and nobody has noticed yet.

What is the customer impact? Stated in the language a customer would use — "checkout fails for about one in five card payments" — not in the language of the monitoring system. Impact drives severity, severity drives who gets woken up, and a wrong impact statement in the first five minutes propagates through every decision that follows.

What are we protecting? Availability, data integrity, or the investigation itself. These conflict. A restart that restores service can destroy the evidence needed to prevent a recurrence, and that trade-off should be made deliberately by the person in command rather than accidentally by whoever reaches the console first.

★ KEY POINTS

- Command is a role, not a rank. The most senior person in the channel is often the worst choice — they are usually the one you want debugging.
- Say role assignments out loud, in the channel, in writing. Assumed roles are unassigned roles.
- A hypothesis is a thing you are trying to disprove, not a thing you are defending.

Exercise 1.1 Opening a live incident

PAIRS · 15 MIN

Your partner reads out the scenario card. Without discussion, write the exact words you would post in the incident channel in your first ninety seconds.

Your opening message — roles, impact, next update time.

What did you deliberately not say, and why?

Exercise 1.2 Impact translation

INDIVIDUAL · 10 MIN

Rewrite each monitoring alert as a one-sentence customer impact statement.

"p99 latency on payments-api exceeded 4000 ms for 5 minutes."

"Replication lag on orders-db-replica-2 is 340 seconds and rising."

Lesson 2 · Communicating while the fire burns

75 minutes

LESSON OBJECTIVES

- Write a status update in under three minutes using a fixed structure.
- Choose an update cadence and hold it even when there is nothing new.
- Distinguish what is known, what is suspected and what is being done.

The cost of silence

Silence during an incident is not neutral. Every minute without an update, the people outside the response fill the gap themselves — the account manager guesses, the executive escalates, and three engineers who are not on the incident join the channel to ask whether it is related to their deploy. Each of those interruptions costs the responders more time than the update would have.

The remedy is a cadence stated up front and then held without exception: "next update at 14:20, whether or not we know more." An update that says "no change, still investigating the payment gateway path, next update 14:40" is a good update. It tells everyone outside that the response is alive and that they do not need to check.

Updates should separate three things that responders habitually blur together: what is **known** (observed and verified), what is **suspected** (the current hypothesis, explicitly labelled), and what is **being done** (the actions in flight and who owns them). Mixing them is how a hypothesis becomes a fact in the retrospective.

Writing for the person who just arrived

Assume every update will be read by someone who has seen none of the previous ones. That means each update carries a one-line restatement of impact, even at update number nine. The responders find this repetitive; the audience does not, because the audience turns over constantly.

Avoid the passive voice for actions and the future tense for hopes. "We are failing over the primary" is an action. "The issue should resolve shortly" is a wish, and when it does not resolve shortly, it is the sentence everyone quotes back.

★ KEY POINTS

- Announce the next update time in every update. Cadence beats content.
- Label hypotheses as hypotheses, every single time.
- Restate impact in every update — your audience is always new.

Exercise 2.1 Three-minute status update

INDIVIDUAL · 10 MIN

Using the scenario from Lesson 1, write update number three. Time yourself; stop at three minutes even if it is unfinished.

Update text.

Exercise 2.2 Known, suspected, doing

TABLE · 8 MIN

Split your update above into its three components. If a line does not fit any column, decide whether it belongs in the update at all.

KNOWN	SUSPECTED	BEING DONE

Exercise 2.3 Cadence audit

GROUP · 12 MIN

Review the transcript excerpt in your handout. Mark every gap longer than your team's stated cadence, then check the boxes below.

- [ ] Impact restated in every update
- [ ] Next update time given in every update
- [ ] Hypotheses explicitly labelled
- [ ] Role assignments visible to a newcomer
- [ ] Resolution announced separately from the all-clear

Lesson 3 · Reviews that change something

105 minutes

LESSON OBJECTIVES

- Build a timeline from evidence rather than from memory.
- Ask "what made this reasonable at the time?" instead of "who missed it?"
- Leave the room with owned, dated, falsifiable actions.

Timeline first, analysis second

A review that begins with analysis produces a story that fits whatever the loudest participant already believed. A review that begins with a timeline — timestamps, log lines, message excerpts, deploy records, with no interpretation attached — produces disagreement early, while it is still cheap to resolve.

Building the timeline is genuinely tedious and always surfaces something unexpected: the alert that fired nine minutes before anyone noticed, the runbook step that was skipped because the link was dead, the second responder who spotted the cause at 14:05 and mentioned it in a thread nobody was reading.

Only when the timeline is agreed does the group move to analysis. The framing question is not "what went wrong" but "what made each action look reasonable to the person taking it?" People do not choose the wrong action; they choose the action that made sense given what they could see, and the fix is almost always in what they could see.

Actions that survive the week

Most review actions die quietly. They die because they are aspirational ("improve monitoring"), unowned ("the platform team should"), or undated. An action worth writing down has a named owner, a date, and a way to tell whether it happened.

Cap the list. Three actions that ship beat eleven that decorate a document. If the group produces more, rank them and let the bottom of the list go — explicitly, in the room, rather than by attrition over the following month.

★ KEY POINTS

- Separate timeline from analysis, in that order, with a visible boundary between them.
- Replace "why did you" with "what did you see".
- Every action gets a name and a date, or it does not get written down.

Exercise 3.1 Reconstructing a timeline

GROUP · 25 MIN

From the evidence pack, write the six events you are most confident about. Timestamp each one and cite where it came from.

Timeline — time, event, source.

Exercise 3.2 Reframing the blame sentence

PAIRS · 10 MIN

Rewrite each sentence so that it asks about the system rather than the person.

"The on-call engineer ignored the first alert."

"Nobody followed the runbook."

Exercise 3.3 Three actions

INDIVIDUAL · 12 MIN

Write three actions from your own team's most recent incident. Each needs an owner, a date and a test for completion.

Action 1 — owner, date, how you will know it is done.

Action 2 — owner, date, how you will know it is done.

Action 3 — owner, date, how you will know it is done.

Notes and commitments

What will you do differently in the next four weeks? Be specific enough that a colleague could check.

PARTICIPANT SIGNATURE

FACILITATOR SIGNATURE

DATE COMPLETED