

Gamedays

How to prepare your teams for real life incidents

Thanos Amoutzias
Software Engineer

What's the avg cost of downtime per minute?

5600\$*

*Source: <https://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime/>

Depends on your business

- Revenue lost
- Productivity
- Data loss
- Brand reputation
- Recovery costs

What's our goal as SREs?

- Minimize downtime
- Handle incidents

How many of you are on-call right now?

To ACK the incident press “4”.

What would you do?

- RTFM?
- Hope?
- Handle the incident!

How do you get your team ready for real incidents?

Gamedays

Staged incidents for engineers to solve.

Where to run those incidents?

Disaster Recovery

“Assume production is broken/deleted.
Create a new environment, spin up your
services, and recover the state from
Sandbox”

Outcome

- Environments to break
- Improved Disaster Recovery Documentation
- Upskilled Engineers
- Improved bootstrapping of the services. One step closer to `tf apply`

Incident Design

- Likelihood
- Learnings
- Set-up

Incident Examples

- Broken API Gateway
- Missing DB Indexes
- Scaling
- Secret/Config Mgmt
- DoS

Outcome

- Upskilled Engineers
- Improved Systems
- Improved Processes

Organization

Teams

- 6-7 engineers per team
- Mixed seniority
- Mixed domain knowledge
- ~1 IRT member in each team

Schedule

- 5-6 incidents for each team in a day
- 1h per incident
- 20' buffer
- 1h lunch break
- 30' intro explaining the whole process
- 30' for wrap-up and feedback form

|Team <> - Disaster Recovery

Goal

Create a new env for your services called "Gamedays".

The new env should:

- use sandbox data
- be reachable via apigee (if applicable)
- new project is up and running

Desired outcome:

- Functional environments that we can break during Chapter 2.
- Create TOX for your findings
- Improve your documentation
- Touch old infrastructure
- Build confidence in the team
- If you already have a Disaster Recovery Plan, update it, if not, time to create one!

<Service Name>

A few words about the domain and the service name.

Timeline

Important timestamps

Time team started:

Time backup recovery started:

Time backup recovery finished:

Service up and running:

Service reachable via Apigee:

Rest of the timeline - note down any important times that will help in understanding how the whole process looked like.

Steps

Note down the steps taken to bring the service up, will be valuable to also tear the env down. This section will also contribute to creating your service disaster recovery plan in the future.

Findings

Note down findings during the process. E.g., the secret is not in IAC, etc.

Gamedays Incident Playbook

Title: <Example: Cloud SQL scalability issue>

| | |
|---|---|
| Facilitator: | <Example: Developer Dave> |
| Estimated Duration: | <Example: 60 Minutes> |
| Desired learnings: | <Example: Scaling up a GCP PostgreSQL database> |
| Type of the incident: | <Example: Scalability issue of PostgreSQL database> |
| | |
| Domain: | <Example: |
| Product: | <Example: |
| Service: | <Example: x-service> |
| Side-effects to other services: | <Example: y-service, z-service> |
| | |
| Stakeholders: | <To make the life of the facilitator easier, note down which stakeholders the team should inform in theory during the incident> |
| | |
| Additional prerequisites: | <Example: database access> |
| Environment for the role play | <Example: gamedays> |
| Automatic or manual alert? | <Will an alert be triggered automatically, or is a manual one required?> |
| If alert is automatic, list the alerts: | <Example: PRIO1 - Alert - x-service> |

TL;DR

- What does an outage mean for your business?
- Identify the environment to run Gamedays and consider a DR scenario
- Design Incidents (likelihood, learnings, set-up)
- Split the teams
- Good luck, have fun
- Follow-up stories for improvements
- Feedback



Thanos Amoutzias

Building the EV charging infrastructure | DevOps | SRE

