

PagerDuty

Managing Vendor Incidents

What do you do when your vendors are down?

Conf42 Incident Management | October 2024

Speaker Intro

MANDI WALLS

DevOps Advocate at PagerDuty

@LNKCHK

mwalls@pagerduty.com

<https://community.pagerduty.com>



The Cloud is Powerful and Convenient



**STORM
AHEAD**

Add functionality you need with little friction
Get the most up to date features with little
need to manually update

It also brings risk



**What matters to your users
should matter to you**

What matters to your users
doesn't make it to your vendors





Organize Your Vendors

Tier 1

Directly **customer** facing or in **support** of **production**

IaaS

Shopping cart, payments

Search

Metrics / monitoring / alerting

An incident on a **Tier 1** provider might mean a **revenue** hit for your organization

Tier 2

Part of your **build** and **test** process

QA tools

CI/CD and Deployment

IaC

Source code management

Possibly user documentation, etc

An incident on a **Tier 2** provider might not impact your customers directly, but could impact your **ability** to **ship bug fixes** and **features**

Tier 3

Additional tools and workflow components

Outside of the workflow to get code to production

POCs, experiments, internal reporting

An incident on a **Tier 3** provider should not impede work on production

Other

Apps you rely on for your own **Incident Response** processes

Conference software

Team **chat**

Internal docs like **wikis** etc

These may just warrant a **note** and a **backup** plan
They are probably owned by **IT**, who might already have a plan and/or **runbook**

Note All Vendors

Know customer behaviors

What features are they engaging with on your products

Which of those features are provided by external vendors

Include vendors in **service diagrams!**

Build Positive Vendor Relationships

For your **Tier 1** vendors:

- Consider buying up to a **premium support package**
- Not for help with implementation, but for access to **VIP support** and **account management**
- Have those regular calls with the vendor - so they know who you are and how important they are to your business
- It helps to be helpful - with bug reports, feature feedback, etc - that gets you more time with the vendor

When your vendor has an incident

Your **customers**
are seeing a
problem
Your team is alerted
by **PagerDuty**

An **incident
response** process
starts

Engineers
investigate

Troubleshooting
points to the
vendor

Check the vendor's **status page**, but there's nothing.

Now what?

Owning the Vendor Relationship

Multiple teams interact with vendors:

- Procurement can be its own discipline
- Legal and finance may be involved before engineering is brought on
- Program managers or project managers for implementation
- Depending on the scope of the product, teams like Platform Engineering or SRE

When you are relying on a **vendor** for your **production environment**, someone or some team who has an **on-call responsibility** should be the **internal contact point** for that vendor.

Vendor Runbooks

Have these for all **Tier 1** and **2** vendors

Available to everyone!

Collect key information:

Account numbers or **user IDs**

Contract information - **featureset** purchased, level of **support**

Email address, support **phone** number, support **website**

Do you have a dedicated **account manager**? Add their info

Status of your account and the **renewal date**

Any guides for vendor-specific reporting, i.e. **error codes**, stack traces

Also good to have when you've caused your own problem - going over purchased limits, bad requests, etc

Contact or Wait?

Very **large** outages are noticeable and might make the **news**

Small subsystems incidents usually don't

Making **contact** with your **account team** might get you on the **first-notified list**

If you decide to **wait** it out, appoint a **contact person** in your organization to keep an eye on the **vendor status**

Use your established **stakeholder communications** plan to keep the rest of your organization up to date

Engage Your Support Team

Update your **status pages** and
customer messaging

Beware **disparaging** comments



Run a Post-Incident Review

Just like for **your own** incidents

Creates a **timeline** and builds **history** for vendor **reliability**

Discuss any **missing** information or **new learnings** about the vendor

Invite the vendor!

Discuss if your organization should find a **new vendor**

After the incident is **concluded**, not during!

Summary

1. **Organize** your vendors
2. **Document** relationship owners
3. Create vendor **runbooks**
4. **Keep in touch** with important vendors
5. Keep your **customers informed**
6. Run **post-incident reviews** after vendor incidents

Learn More

<https://pagerduty.com>

<https://response.pagerduty.com>

<https://www.pageittothelimit.com/vendor-incidents-with-jeff-martens/>

Join our community at <https://community.pagerduty.com>!

