



# TALES AFTER MIDNIGHT

*FOR WHOM THE PAGER TOLLS*



# TALES AFTER MIDNIGHT



*FOR WHOM THE PAGER TOLLS*

*AN ANTHOLOGY OF SHORT CAUTIONARY HORROR STORIES*



@CEDMAX

# WEBMASTER BEFORE IT WAS COOL

TECH LEAD  
CONDÉ NAST INTERNATIONAL

# TALES AFTER MIDNIGHT

*FOR WHOM THE PAGER TOLLS*

*"THERE ARE 2 TYPES OF ENGINEERS:  
THOSE THAT HAVE BROKEN PRODUCTION  
AND THOSE WHO ARE ABOUT  
TO BREAK PRODUCTION"*

*MATT BOYLE*

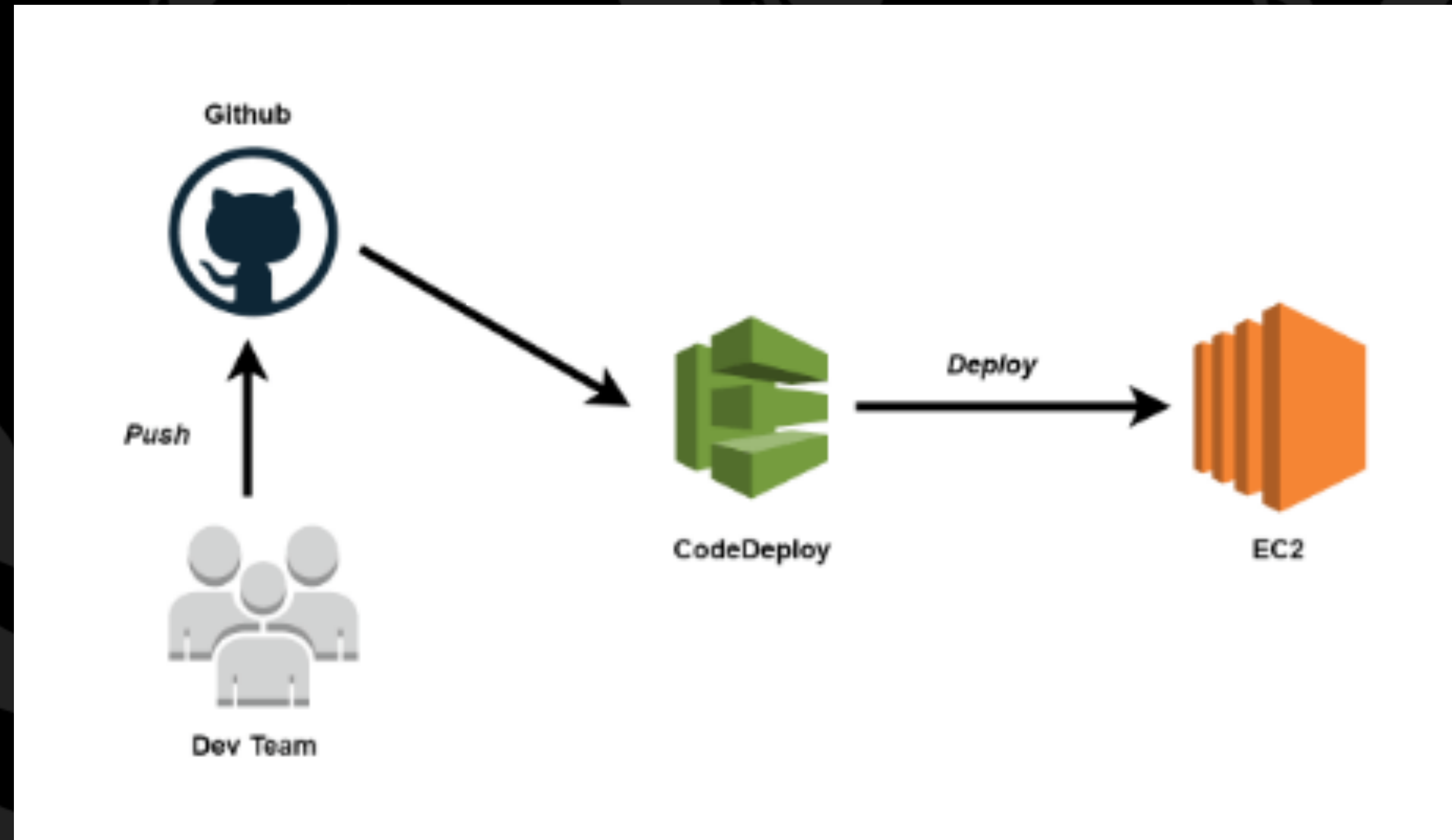
CHAPTER #1

# DON'T DEPLOY THAT CODE









**AWS** CodeDeploy

DON'T DEPLOY THAT CODE



**DON'T DEPLOY THAT CODE**



<b>Application*</b>	HollowWebApp  
<b>Compute platform</b>	EC2/On-premises
<b>Deployment group*</b>	HollowWebApp 
<b>Deployment type</b>	In-place 
<b>Repository type*</b>	<input checked="" type="radio"/> My application is stored in Amazon S3 <input type="radio"/> My application is stored in GitHub
<b>Revision location*</b>	s3://hollow-deploy/AppDeploy1.zip?etag=32809a5!  

DON'T DEPLOY THAT CODE



**DON'T DEPLOY THAT CODE**









**DON'T DEPLOY THAT CODE**



**6:00**

**DON'T DEPLOY THAT CODE**

<b>Application*</b>	HollowWebApp  
<b>Compute platform</b>	EC2/On-premises
<b>Deployment group*</b>	HollowWebApp 
<b>Deployment type</b>	In-place 
<b>Repository type*</b>	<input checked="" type="radio"/> My application is stored in Amazon S3 <input type="radio"/> My application is stored in GitHub
<b>Revision location*</b>	s3://hollow-deploy/AppDeploy1.zip?etag=32809a5!  

DON'T DEPLOY THAT CODE

 WHAT WENT WRONG

- THE PROCESS WAS PRONE TO ERRORS

 WHAT WENT WELL

- OUR ROLLBACK STRATEGY

 WHAT WE LEARNT

- AUTOMATE ALL THE THINGS

DON'T DEPLOY THAT CODE

CHAPTER #2

# THE CSS OF DOOM



THE CSS OF DOOM





Firefox, Opéra, IE 7 et + ...



IE6 et -



```
img {  
  behavior: url("pngbehavior.htc");  
}
```

```
var supported = (/MSIE (5\.5)|[6789]/  
  .test(navigator.userAgent)  
  && navigator.platform == "Win32");
```

```
var blankSrc = "blank.gif";
```

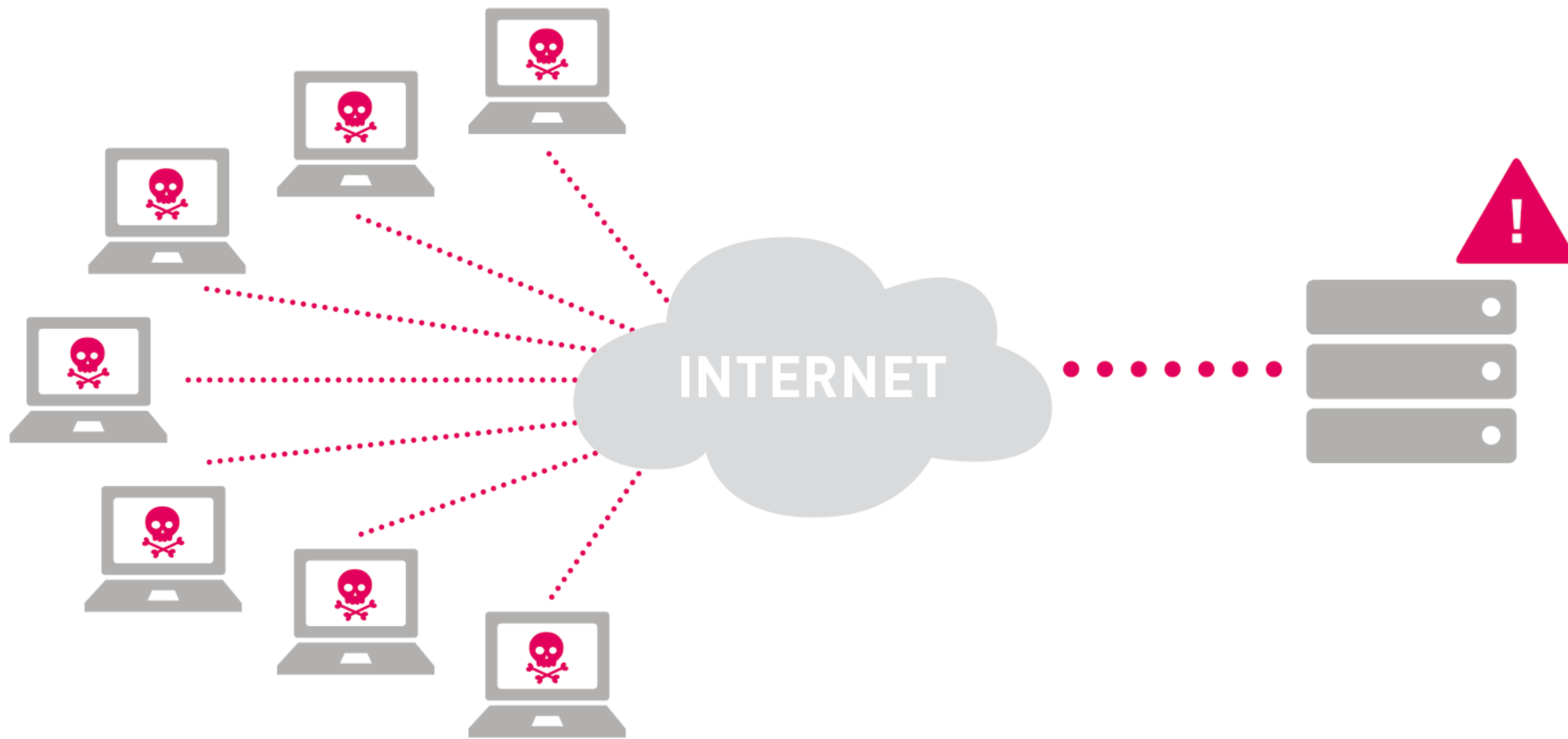
```
if (supported) fixImage();
```

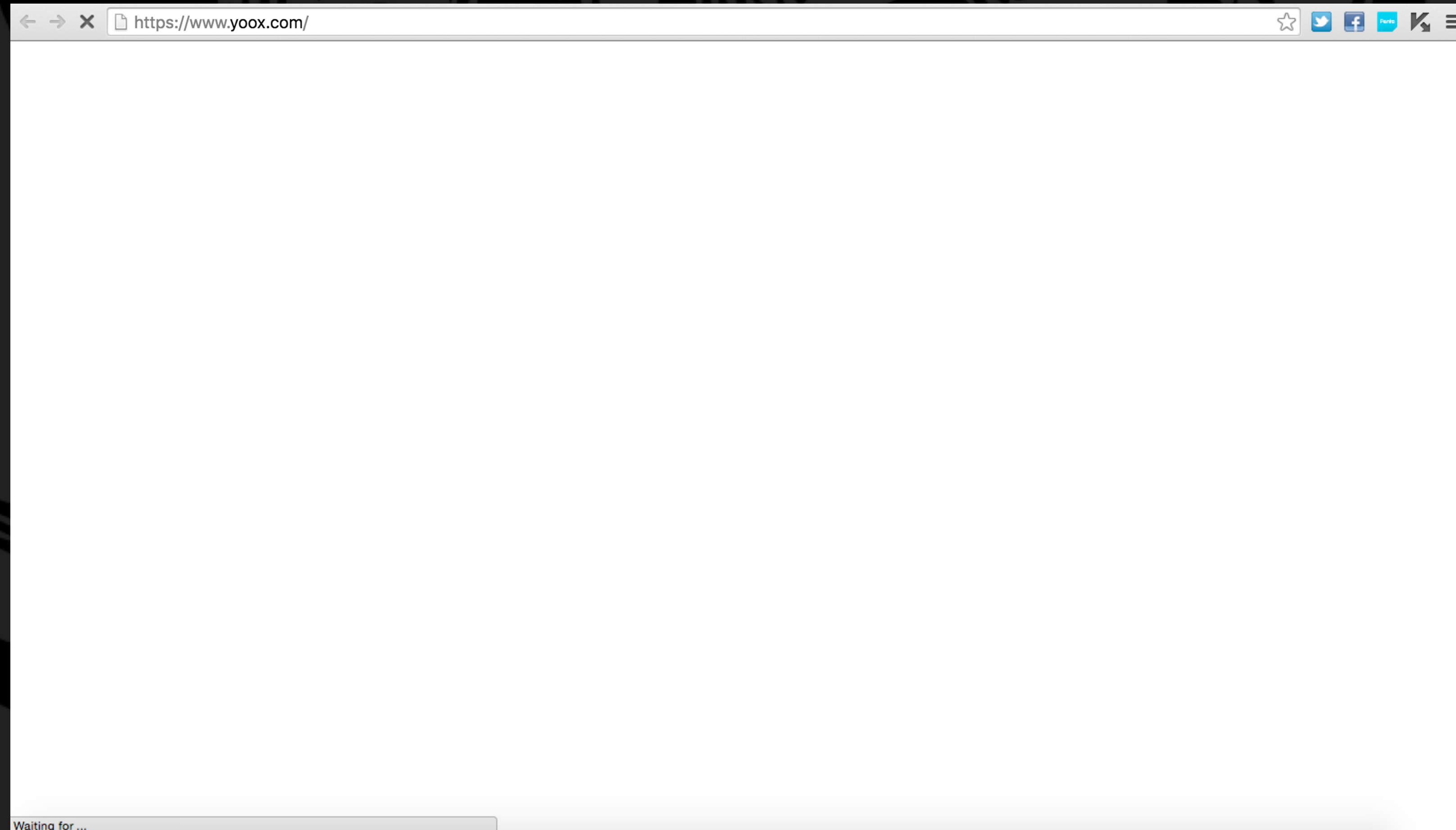
```
function fixImage() {  
    var src = element.src;  
  
    if ( /\.png$/ .test( src.toLowerCase() ) ) {  
        element.src = blankSrc;  
        element.runtimeStyle.filter =  
            "progid:DXImageTransform.Microsoft.AlphaImageLoader(src='"  
            + src + "',sizingMethod='scale')";  
    }  
}
```

```
var supported = (/MSIE (5\.5)|[6789]/  
  .test(navigator.userAgent)  
  && navigator.platform == "Win32");
```

```
var blankSrc = "blank.gif";
```

```
if (supported) fixImage();
```







## 💀 WHAT WENT WRONG

- I DON'T EVEN KNOW WHERE TO START:
  - USING THE PROD DB FOR MONITORING
  - ASSUMING THE CSS CAN'T BREAK ANYTHING
  - SILOED & BLAME CULTURE

## 🧘 WHAT WENT WELL

- I HAVE A FUN STORY TO TELL
- OUR ROLLBACK STRATEGY

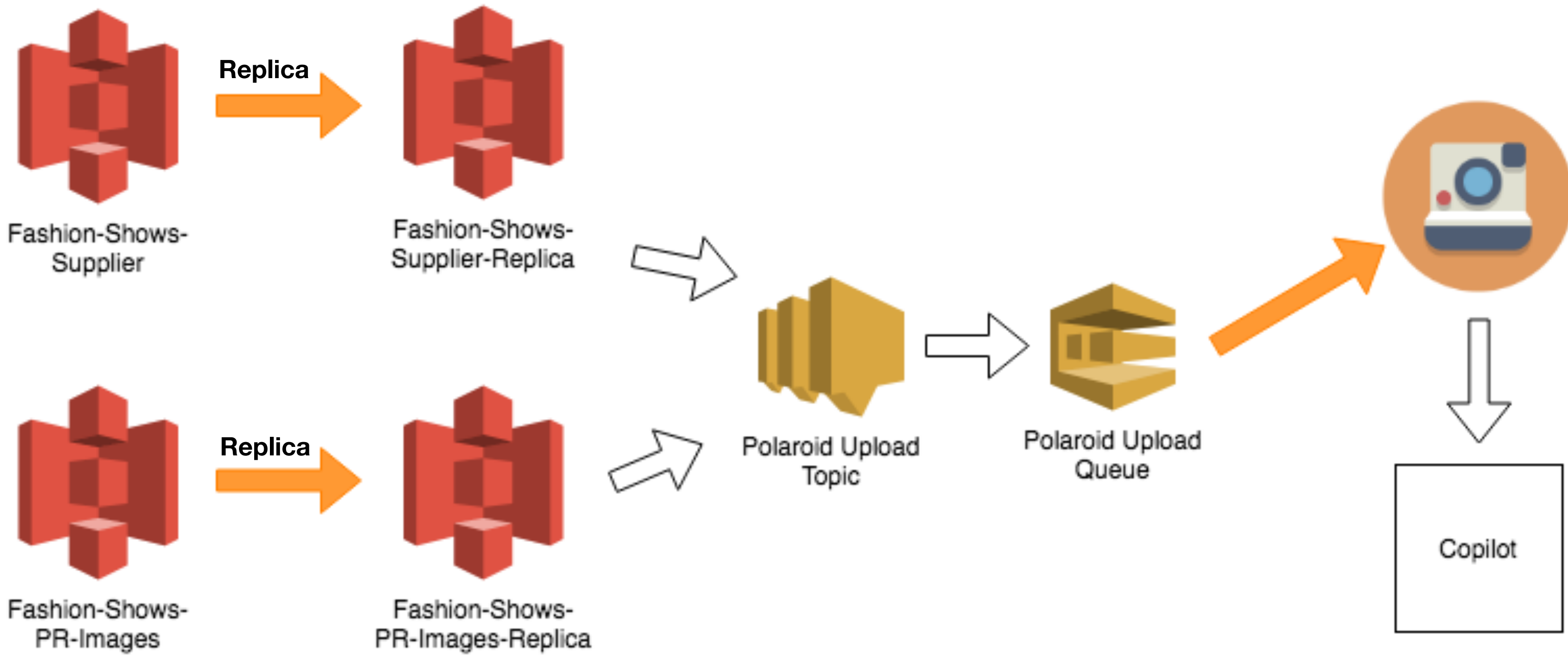
## 🧐 WHAT I LEARNT

- YOU NEED TO UNDERSTAND THE SYSTEM

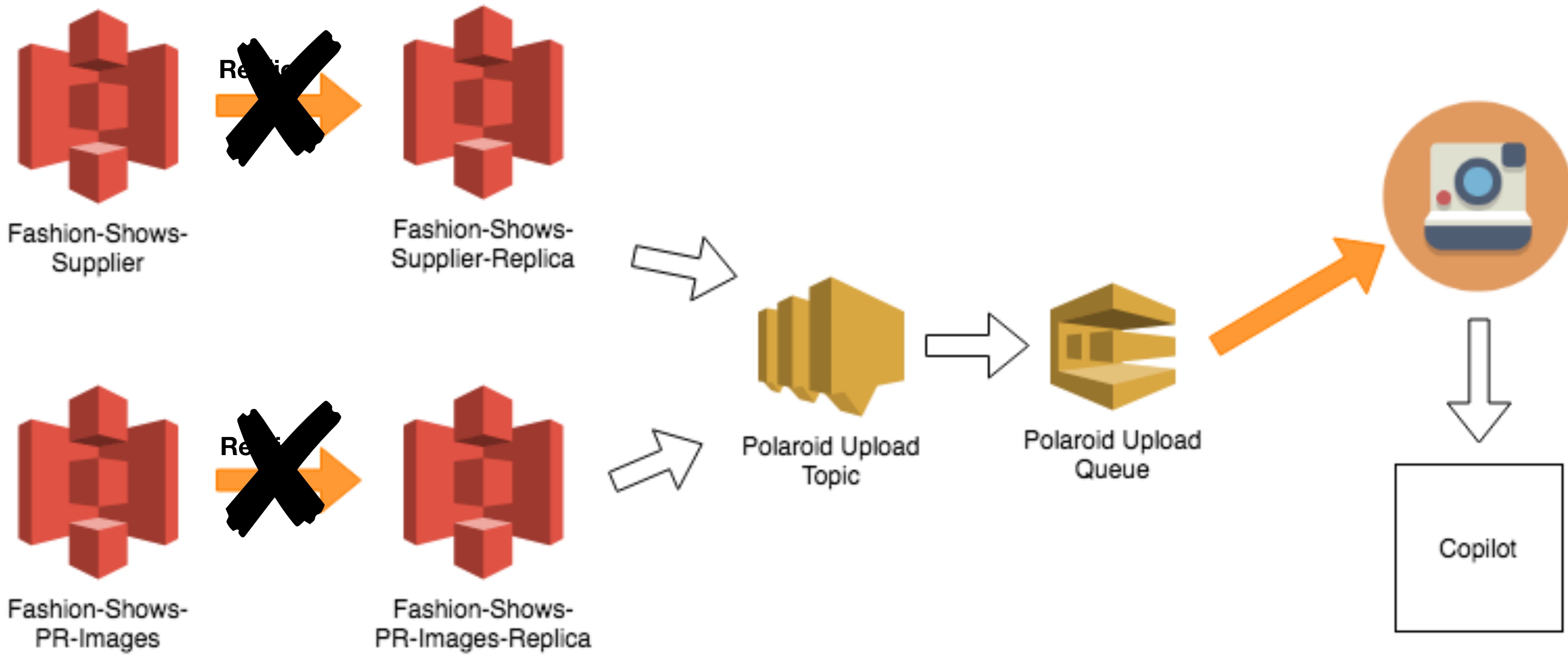
CHAPTER #3

# POLAROID

# Polaroid



# Polaroid





2 DAYS

POLAROID



POLAROID



**RUN BOOKS DIDN'T HAVE ANYTHING ON THE ISSUE  
THE SUPPORT ESCALATION PATHS WERE WRONG  
WE FAILED TO COMMUNICATE THE FIX**

## WHAT WENT WRONG

- MONITORING WASN'T SETUP FOR THE REPLICAS
- THE RUNBOOKS WEREN'T UPDATED

## WHAT WENT WELL

- WE CAUGHT THE PROBLEM BEFORE IT BECAME CRITICAL

## WHAT WE LEARNT

- YOU CAN'T CONTROL EVERYTHING
- COMMUNICATION IS KEY





APPENDIX

# POST MORTEM

# THE DOCUMENT

## SUMMARY

### Summary

*During Stockholm Fashion Week on Thursday August 30th, image ingestion for fashion show imagery stopped working.*

*Images were present in the fashion show S3 folders but they were not successfully copied into the replica buckets. This appears to have been due to some changes introduced as part of another replica work on other buckets, to support Coco Le Bot.*

# THE DOCUMENT

SUMMARY

TIMELINE

## Timeline

*The replication was affected by the configuration change at **10:30 on the 30th.***

*This issue affected users from **August 30th 10:30 - September 2nd.***

*It was not detected until **August 31st***

### **August 31st**

**4:34,** ██████████ raised a support query with the SRE team, first via email and then followed up on Slack (#sre-support channel)

**14:37** ██████████ alerted the #sre-support channel about an Urgent Fashion Shows issue stating the Selam Fessahaye Show was not ingested into CoPilot.

**14:42** A subsequent email was sent at to the team FAO ██████████ and ██████████. Sanya attempted to access the SRE bucket but had access issues. The support query was responded to instantly on Slack.

**14:47** ██████ replied on the email support request, when prompted by ██████. Our US stakeholders were in cc on the email so they need this notification, after prompting.

**14:48** ██████ asked ██████████ for the URLs for the new polaroid and a runbook of sorts so as to access the new service. There was no documentation but ██████ provided the URLs.

POST MORTEM

# THE DOCUMENT

- SUMMARY

- TIMELINE

- IMPACT

## Impact

*Images for 12 fashion shows (roughly all shows as part of Stockholm Fashion Week) were not successfully copied into the replica buckets, and thus were not ingested into Copilot US via Polaroid.*

*This issue affected the CNI Fashion Shows editorial team, who are responsible for supplying fashion show data, and the Vogue Runway production team who are responsible for publishing that data on Vogue Runway. Two members of the Fashion Shows editorial team had to work over the weekend to manually input the data into Copilot [REDACTED] had to monitor the images coming in and make sure they were sent to the US via [REDACTED] as the automated uploading/ingestion process was disturbed. The Vogue Runway team were unable to publish the shows. Revenue was not impacted.*

# THE DOCUMENT

SUMMARY

TIMELINE

IMPACT

DIAGNOSIS

## Diagnosis

<See the [five whys](#) for an explanation of how to fill in the five whys>

### What happened?

*The s3 bucket where Polaroid reads images from wasn't receiving the images*

### Why did the bucket stop receiving images?

*Configuration changes on the receiving bucket changed the policy disallowing the fashion-shows/pr account from writing to the production bucket. [ [ticket](#) ] The ticket prescribed a solution to the problem.*

### Why were the configuration changes made?

*The change was requested in order to enable Coco Le Bot to access the images.  
S3 Bucket replication only allows single source and destination. Therefore these changes in the policy stopped it functioning. Which was unknown at the time of requesting/making the changes.*

*These changes went undetected for over 24 hours as there currently isn't monitoring in place to detect S3 replica failures.*

### Why were changes made in production and not in staging?

*Inconsistent knowledge about the applications stack, environments and workflows across the teams.*

# THE DOCUMENT

▸ SUMMARY

▸ TIMELINE

▸ IMPACT

▸ DIAGNOSIS

▸ ACTIONS

- **Support model (mid term, being worked on):** Clearly defined, documented and communicated support model; detailing key roles and responsibilities, key steps, tools and comms in the escalation process for P1 issues for engineers, incident managers, product teams, leadership and stakeholders as well as devops training for engineers. Clearer ownership of these tasks between SRE, Infrastructure and product devs
- **Monitoring + alarms (short term mid effort, can go in the backlog):** components monitoring (datadog uploads vs queues, checking the dead letter queue length). Establish what monitoring+alerting solutions we can use for Polaroid and implement them.
- **Dashboard (short term low effort, can be started straight away):** creating a dashboard on datadog to give visibility to the asw clues on the process statuses.
- **Prod readiness (monitoring + Alerting medium/long term):** Define minimum standards around monitoring and alerting, and integrate them into the launch readiness process for deployment of new products or features.

POST MORTEM

## SUGGESTIONS

- KEEP THE MEETING SMALL
- KEEP IT TECHNICAL
- FOCUS ON THE WHYS?
- MAKE SURE THE ACTIONS ARE REALISTIC
- KEEP IT BLAMELESS

# TALES AFTER MIDNIGHT

FOR WHOM THE PAGER TOLLS

## TAKEAWAYS

- SH\*T WILL GO WRONG, BE PREPARED
- KEEP YOUR DOCS UPDATED
- TRY TO UNDERSTAND YOUR APP/STACK
- DON'T BLAME PEOPLE BUT PROCESSES

NGC