

Modern workflows with container-based infrastructures

LATA Conference 2020



Florian Loretan
CTO
Wunder

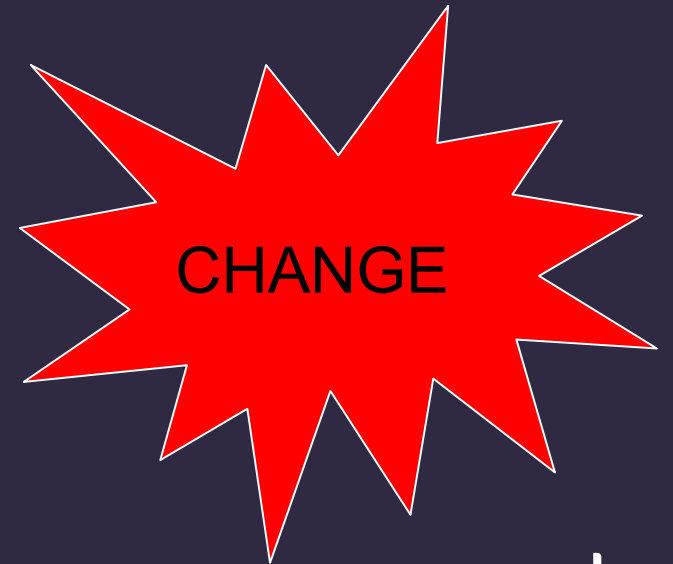


Gatis Špats
DevOps
Engineer
Wunder



1. Less changes = less worries?
2. How to enable change
3. Increase resilience
4. Improving efficiency
5. Conclusion

- Systems can be:
Complex
Unique
Low automation level



Lead group / low performers

106
TIMES FASTER

lead time from
commit to deploy



208
TIMES MORE

frequent code deployments

How many deploys / day or week you have in your information systems?

Do you measure what is the time from feature request or commit till delivery of that change in production?

Could these metrics be better and how?

Enabling change

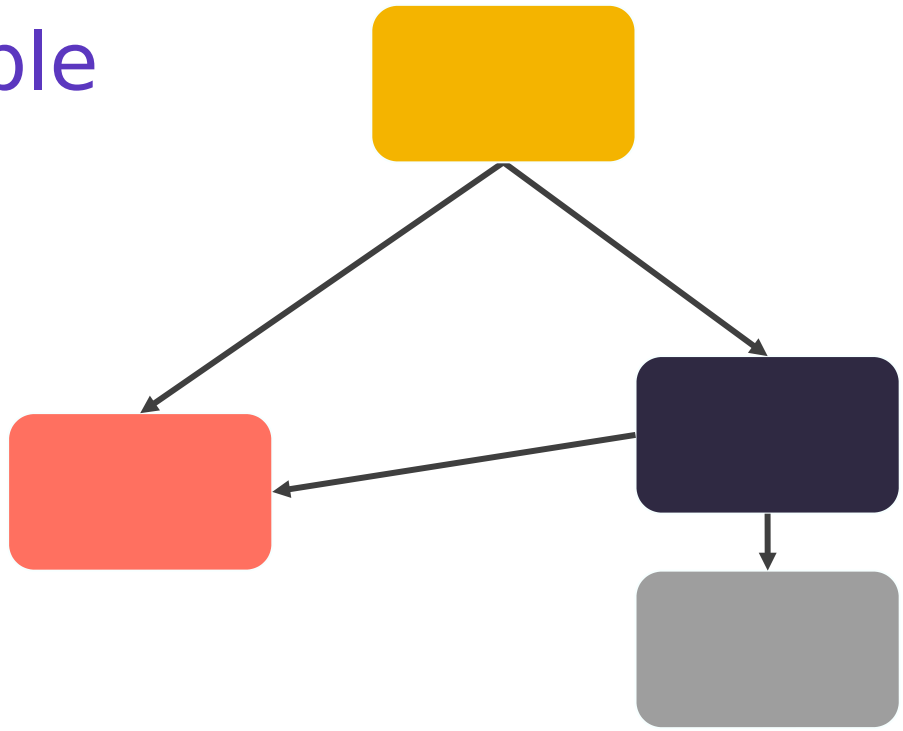


wunder

*Change is risky → Reduce
frequency*

Change often → Reduce risk

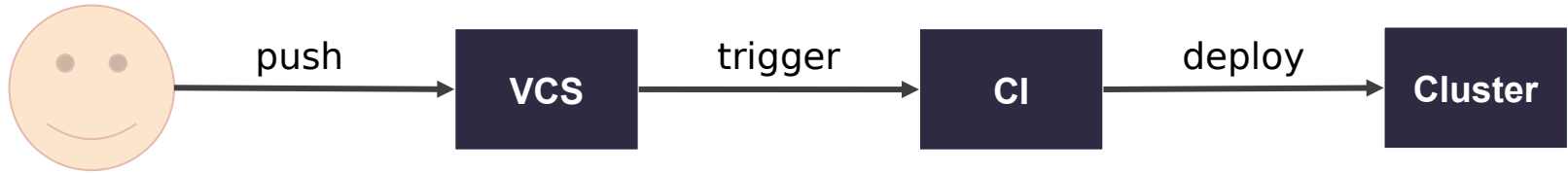
Decouple





Automate

How we do it



- Code
- Tests
- Build instructions
- Deployment instructions
- Access Control

- Validate
- Build
- Push

- Provision
- Execute
- Monitor

Cluster

Project ⇒ Namespace

Branch ⇒ Release

Ingress

Service

Container

Volumes

Branch ⇒ Release

Ingress

Service

Container

Volumes

*13 minutes from nothing to
production*

Increasing resilience



wunder

Reduce people dependencies

- Less manual steps
 - ⇒ less potential for mistakes
- All configuration under version control
 - ⇒ active documentation
 - ⇒ not in people's head
- Automated access control
 - ⇒ less waiting to be privileged
 - ⇒ better security



Observe

Cluster : [REDACTED]

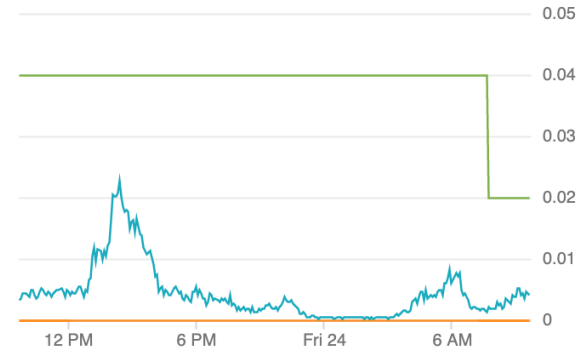
Namespace : [REDACTED]

Pod : production-node-65f77b7b68-r2p79

Error: Readiness probe failed: dial tcp [REDACTED] connect: connection refused

✔ Login to the docker registry	00:00
⌵ Build nginx docker image	00:02
⌵ Build php docker image	00:03
⌵ Build shell docker image	00:03
⌵ Set release name	00:00
⌵ Build local helm chart	00:07
⌵ Dry-run helm install	00:11
⌵ Clean up failed Helm releases	00:01

```
❌ Deploy helm release 00:14  
  
Error: UPGRADE FAILED: cannot patch "test-cluster-ssh-keys" with kind PersistentVolumeClaim: PersistentVolumeClaim  
"test-cluster-ssh-keys" is invalid: spec: Forbidden: is immutable after creation except resources.requests for bound  
claims  
  
Exited with code exit status 1
```



● Used: ● Requested:



Feed Back

- Fix errors once
- Increase speed
- Improve usability

Practice Resilience

- Failures are detected and handled automatically
- Make them invisible to end users
- Practice them

Scale

- Scaling should be handled automatically
- Better resource sharing leads to better scaling
- Treat scaling as a regular thing
- Scale up, but also down

Improving efficiency



wunder

Saving on infrastructure costs

- Better resource sharing leads to less computing resources needed
- Cluster abstraction means easier portability
- Cloud infrastructure means you pay only what you use
- Take the cost of tools into consideration

Improve the productivity of people

- Clear responsibilities leads to better flow
- Independence leads to improved ownership
- Focus on collaboration, not dependencies



Reduce vendor lock-in

- Decoupled components can be managed by different vendors
- Standard APIs improve replaceability
- Higher-level infrastructure abstraction improves portability

Conclusion



wunder

- Practice change
Embrace failure
Improve people processes first



Thank you

:-)



Shaping the Digital
Experience
wunder.io