Welcome to LIGHTNING TALKS

SRE CON®

ASIA PACIFIC

Singapore
June 14
2019

The Rules

- 5 Minutes Per Speaker!
- 15 seconds per slide, auto advancing!
- 2 handheld mics in a Blue/Green deployment
- Speakers be ready to run on stage!

GO!!

SRE CON ASIA PACIFIC

Developing effective project plans for SRE interns

Andrew Ryan <andrewr@fb.com>

Production Engineer, Facebook Inc.

USENIX SREcon Asia, June 2019

So you’re getting interns?

https://pixabay.com/photos/kittens-small-hands-hold-grass-4020199/

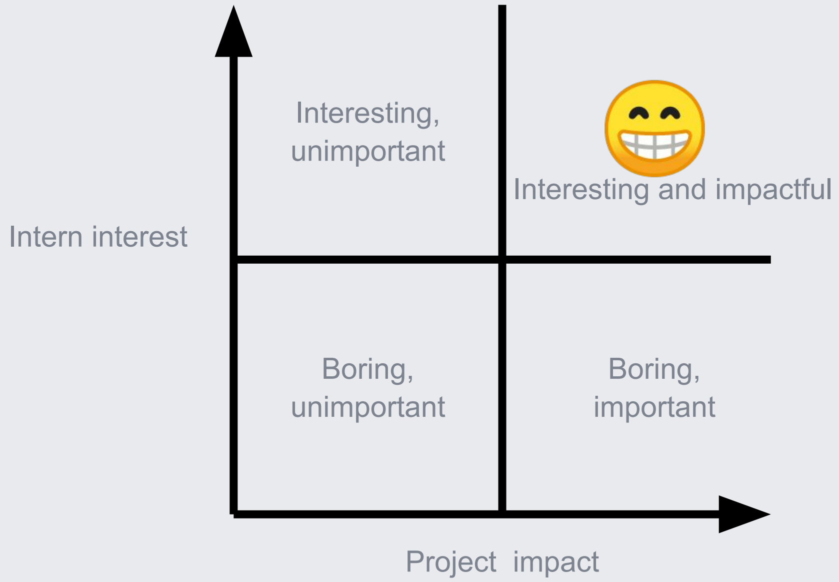
What makes a good SRE intern project?

- Something experimental
  - “Rewrite an existing service in Rust”
  - If it doesn’t work, well, you still have the old one

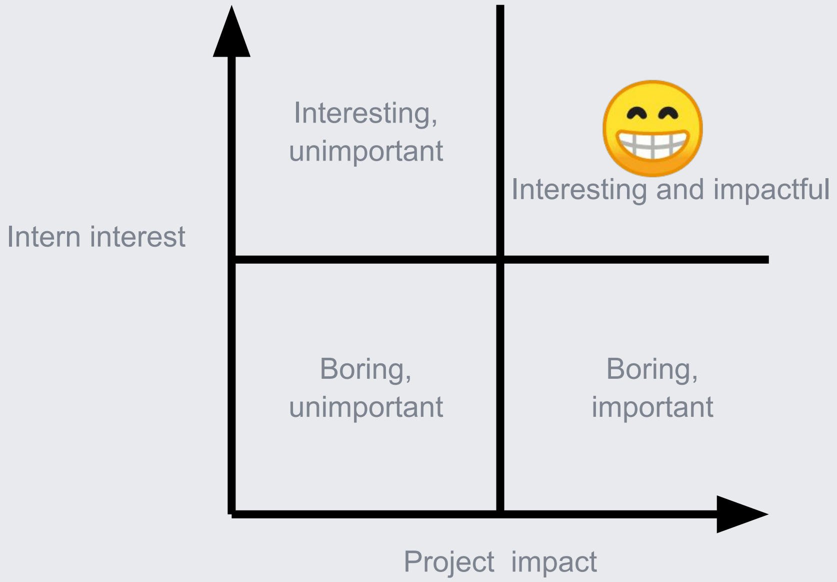
What makes a good SRE intern project?

- Something experimental
  - “Rewrite an existing service in Rust”
  - If it doesn’t work, well, you still have the old one
- Something greenfield
  - “A new service to generate and analyze traceroute data”
  - If it doesn’t work, well, you never had it anyway 😂

Try to maximize intern interest & project impact

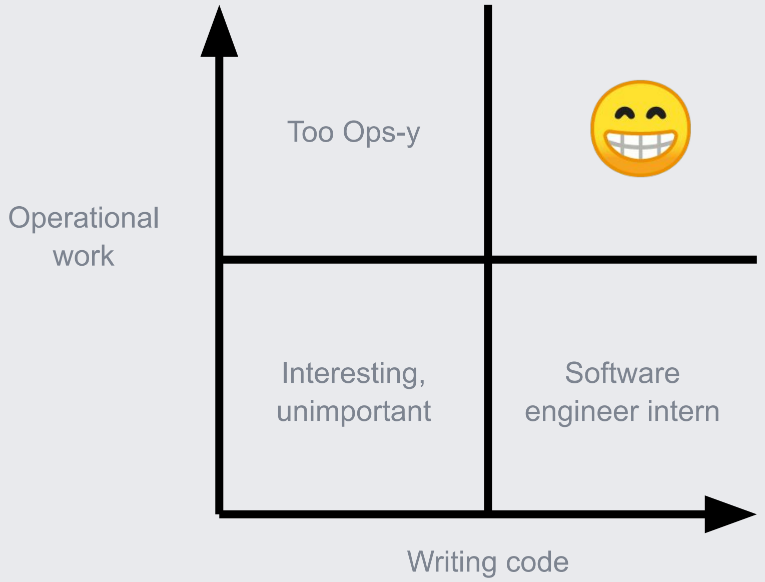


Try to maximize intern interest & project impact

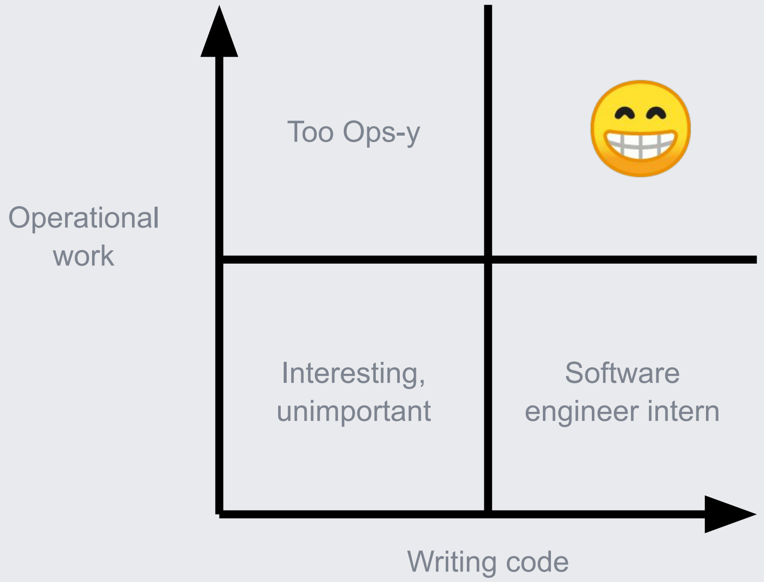


Make your interns ship *something*!

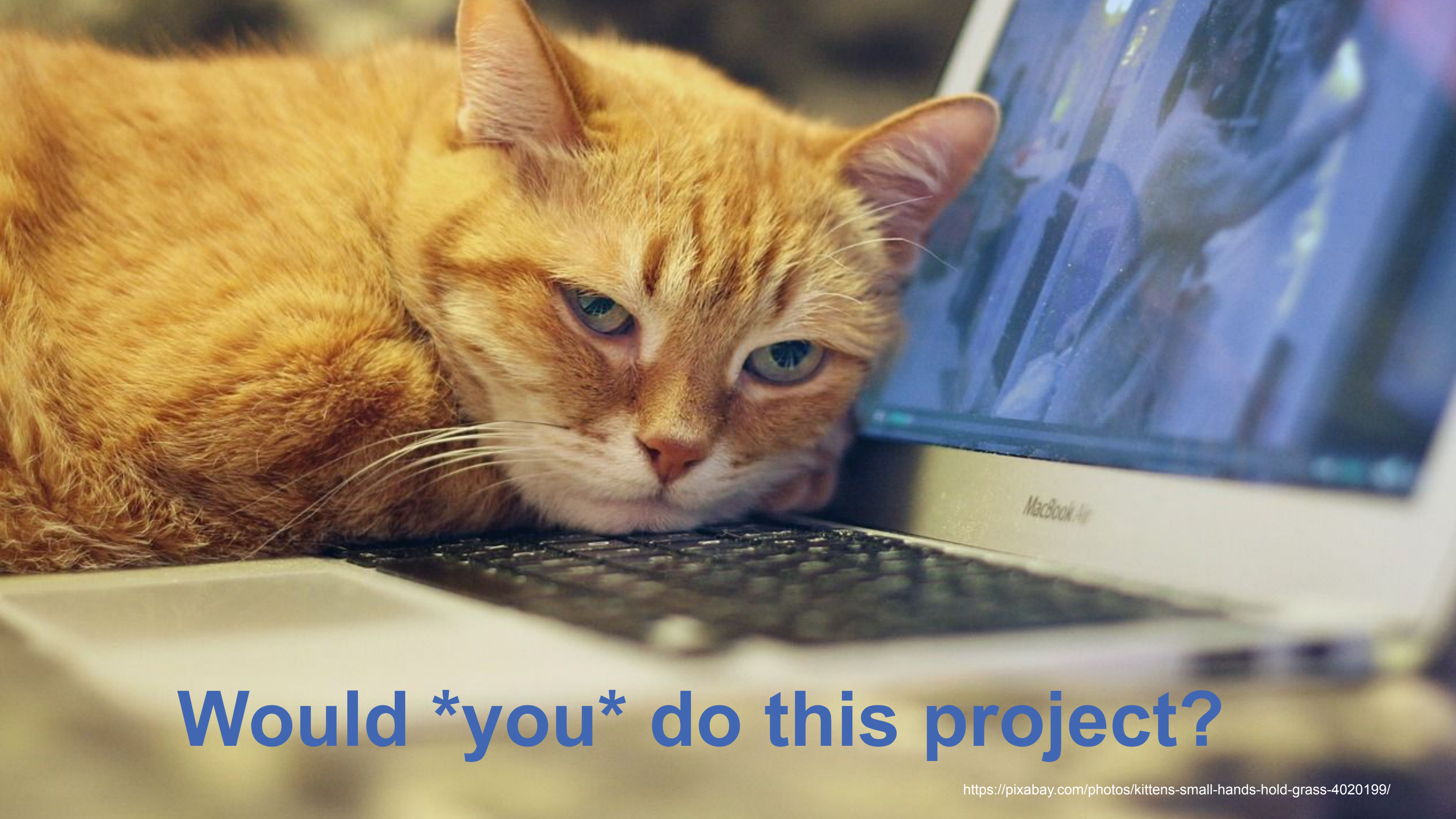
Combine code and operations



Combine code and operations



Give interns a real sense of SRE work



Would *you* do this project?

https://pixabay.com/photos/kittens-small-hands-hold-grass-4020199/

#1: Title and Description

- Be clear and concise
- GOOD: “Improve reliability and speed of initial Chef runs”
- BAD: “Work on initial Chef runs”

#2: Skill requirements

- “Project requires Python, C++, and understanding of BGP”
- Great way to match interested interns with your project!

#2: Skill requirements

- This can be what interns LEARN, not just what they already KNOW
- Interns can learn QUICKLY, especially when motivated by interest!

#3: Risks

- What could go wrong?
  - External technology/team risk
  - Project more complex than you thought

#3: Risks

- What could go wrong?
  - External technology/team risk
  - Project more complex than you thought
- And how will you deal with it?
  - Alternate tech solutions
  - Change project direction/scope

#4: Milestones

- Break up work into SPECIFIC deliverables 1-2 weeks apart
  - Week 2: Complete design and spec of new Foobar service
  - Week 4: Have prototype built with basic functionality

#4: Milestones

- Break up work into SPECIFIC deliverables 1-2 weeks apart
  - Week 2: Complete design and spec of new Foobar service
  - Week 4: Have prototype built with basic functionality
  - Week 6: Take 1% of production requests with new service
  - Week 8: Ramp up to 20% production requests, add dashboards and monitoring

#5: Project Extensions/Minimums

- What if your intern makes more (or less) progress than anticipated?
- Or if the project turns out to be harder (or easier) than you anticipated?

#5: Project Extensions/Minimums

- What if your intern makes more (or less) progress than anticipated?
- Or if the project turns out to be harder (or easier) than you anticipated?
- Think of project extensions and independent “mini-projects”

To recap…

1. Have a written plan before the intern arrives
2. Get the plan peer-reviewed
3. Make the intern ship “something”
4. Review results, repeat next year

Thank you for helping to grow the next generation of SRE’s!

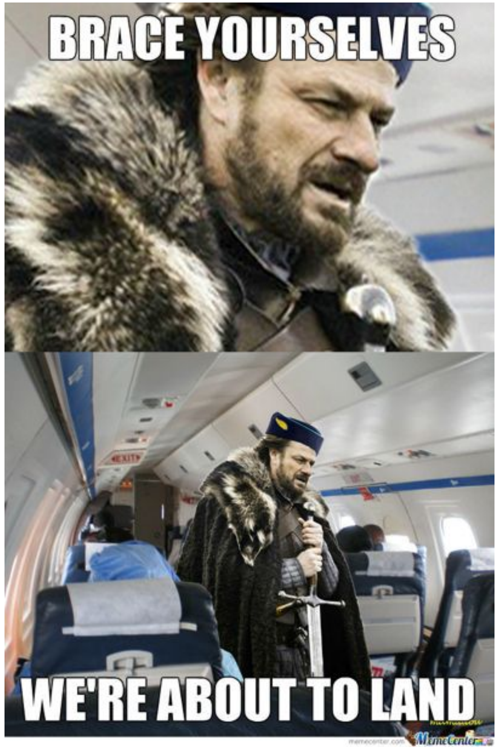
https://pixabay.com/photos/cat-animal-kitten-cute-floor-eyes-2590464/

Zero Downtime cross cluster migration of microservices in Kubernetes

VMware GitHub Project: https://github.com/vmware/k8s-endpoints-sync-controller

Migration of People Vs. Migration of Services

- On planes
- Gives me anxiety



@memecenter.com

- Using our solution
- Gives me anxiety++

Maintaining SLA of services during migration is tough.

I had a dream last night, the plane went down. You were in it.

@giphy.com

Smooth transition between two clusters with minimal risk is tougher.

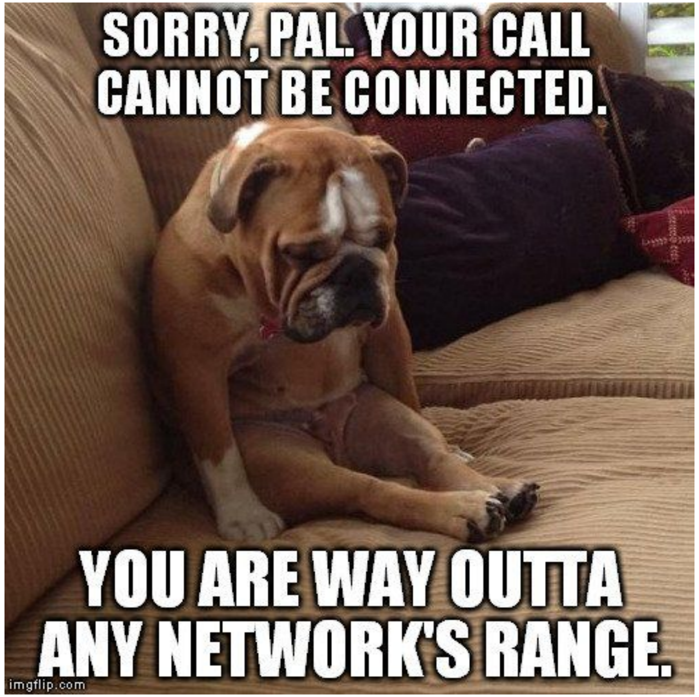
You don't know the definition of anxiety until you look out a plane window and see this.

@me.me

The clusters can be spread across multiple regions and multiple cloud providers – it shouldn't be a problem!

@imgflip.com

A service should be able to talk to services in other clusters in the same way that services communicate within a single cluster


SORRY, PAL. YOUR CALL CANNOT BE CONNECTED.
YOU ARE WAY OUTTA ANY NETWORK'S RANGE.
imgflip.com

@imgflip.com

Migrating any service- whether it is stateless or stateful should happen seamlessly

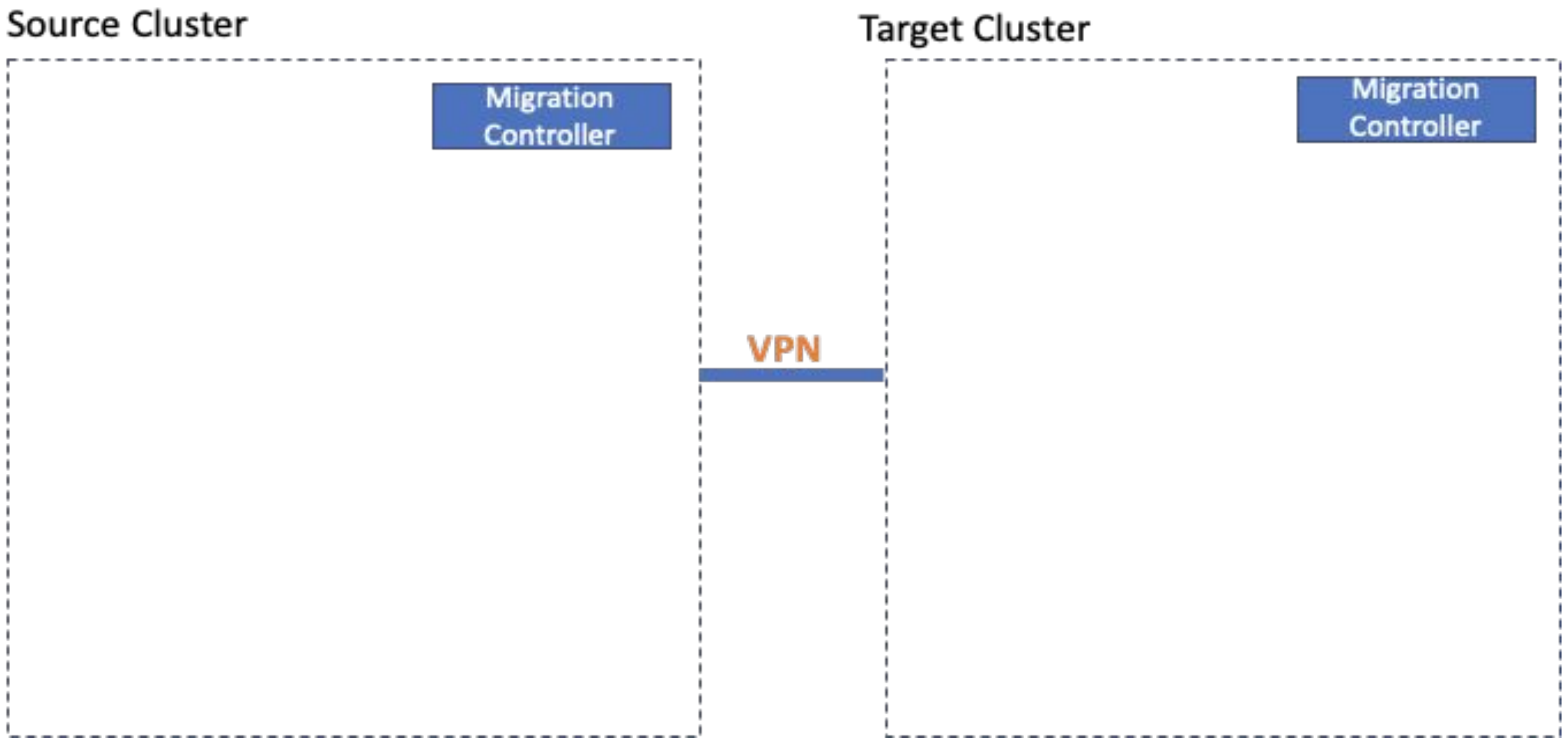
I asked for a wing seat & a discount
Next time I'll be a bit more specific
imgflip.com

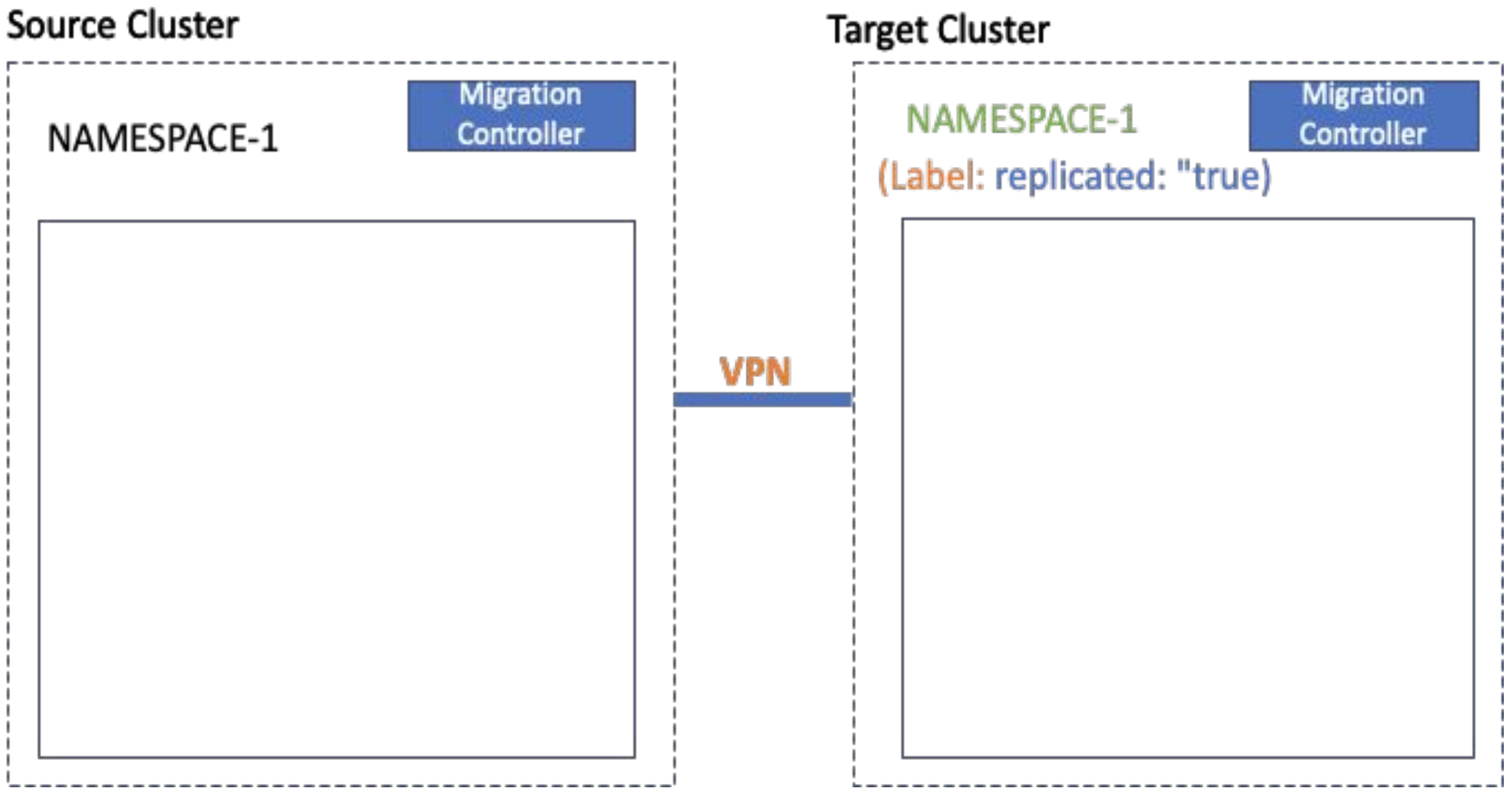
@imgflip.com

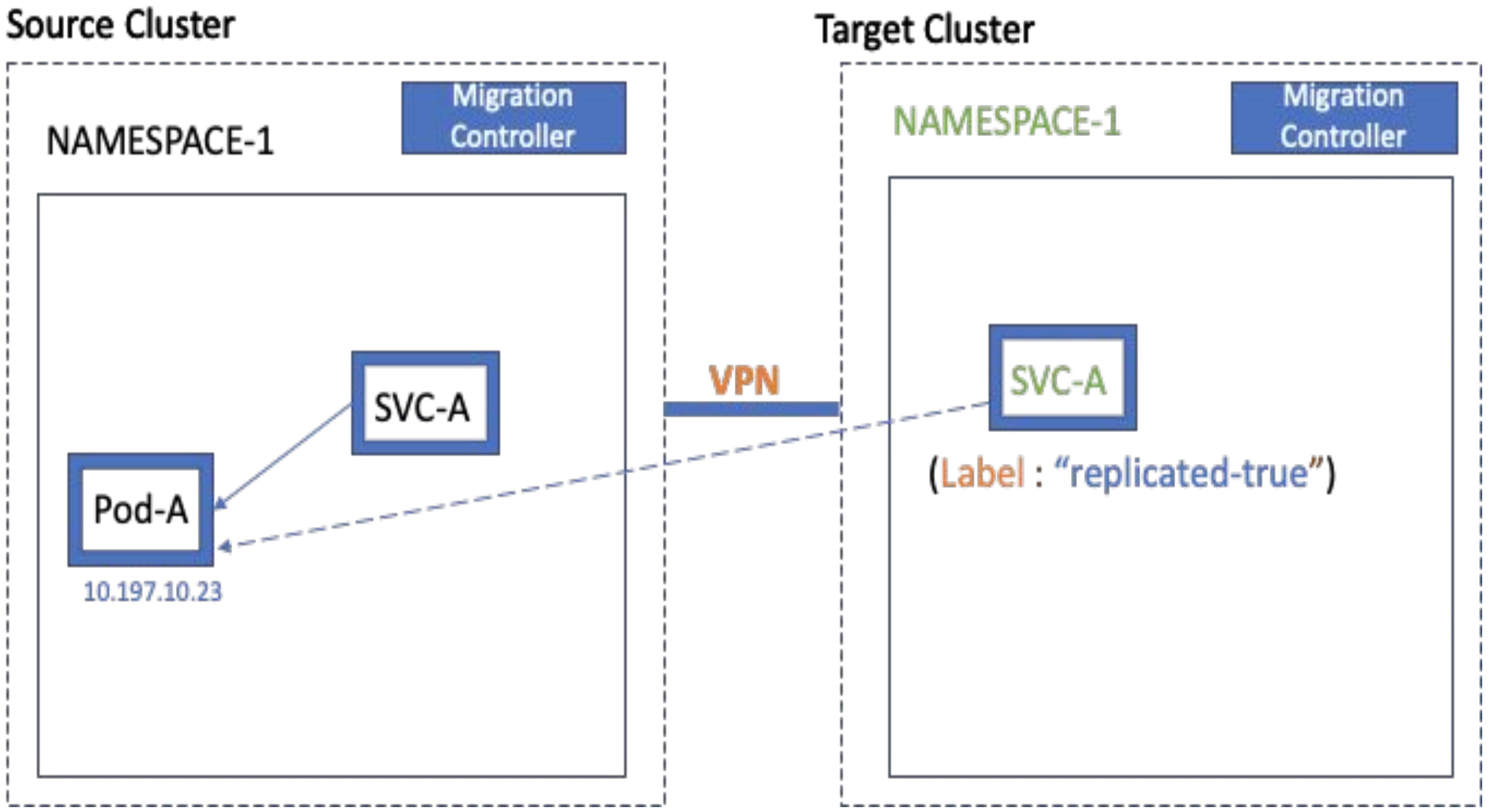
@imgur.com

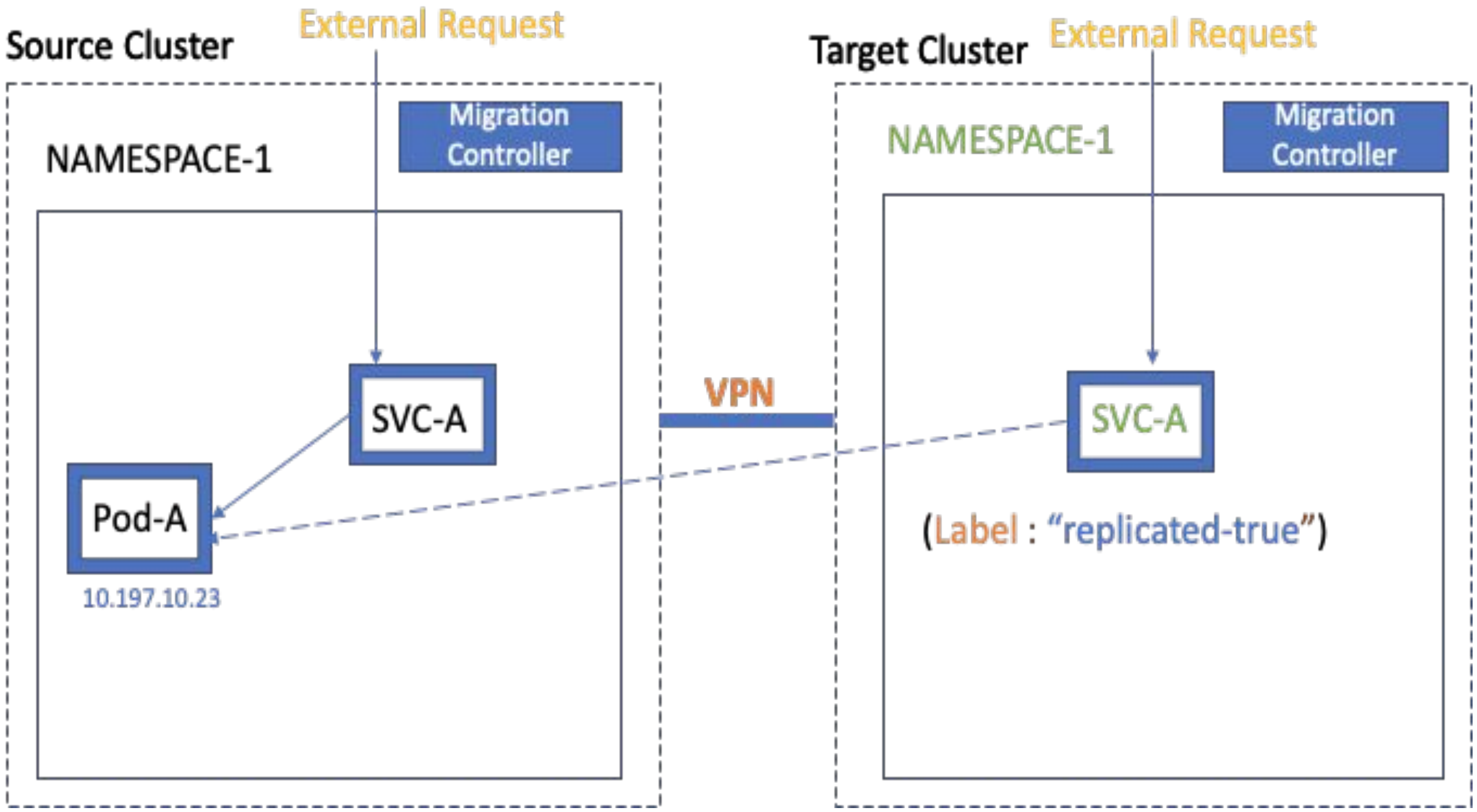
Now that you have understood our problem statement, and our obsession with memes...

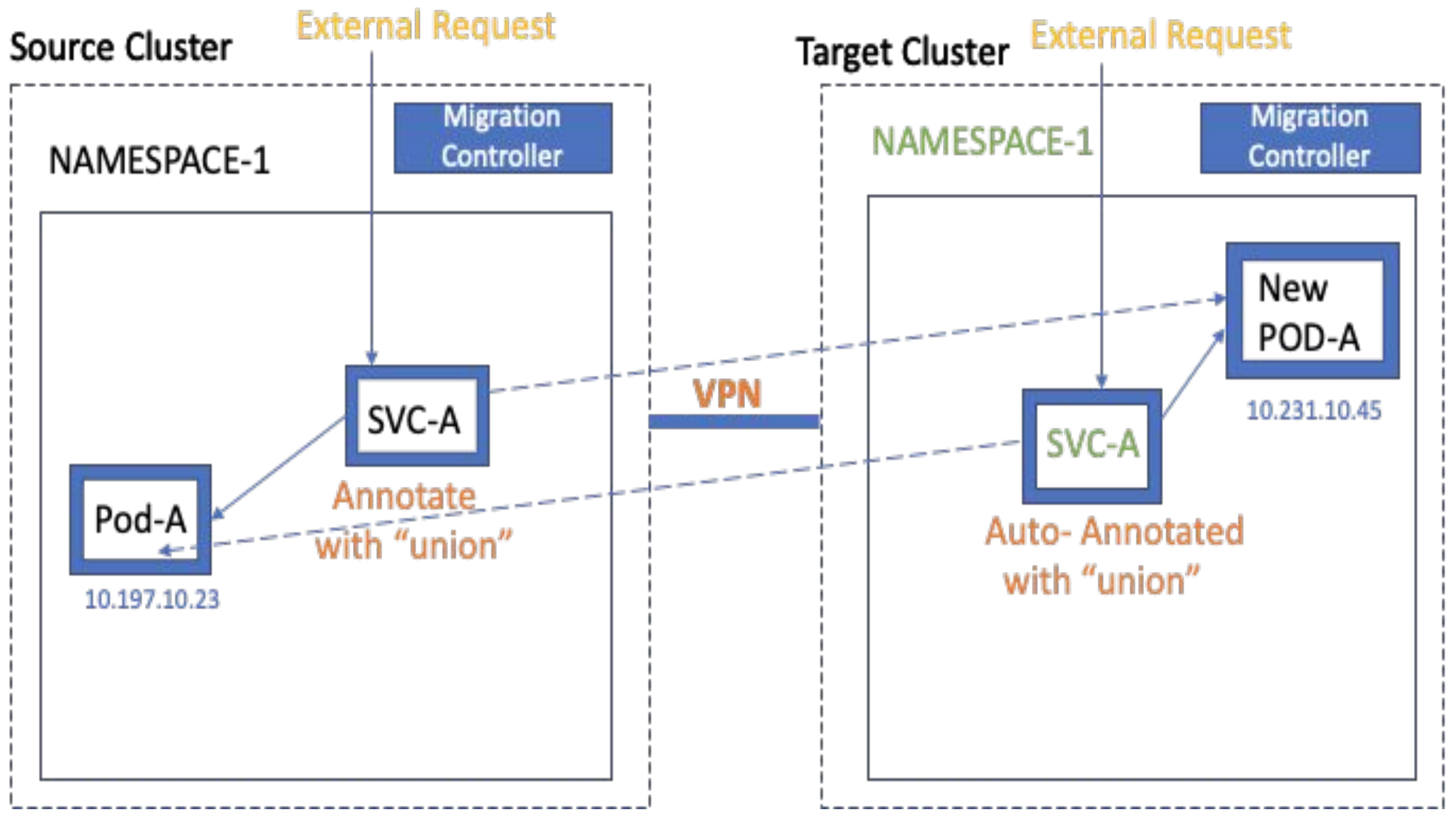
How did we achieve migration of services between Kubernetes clusters?

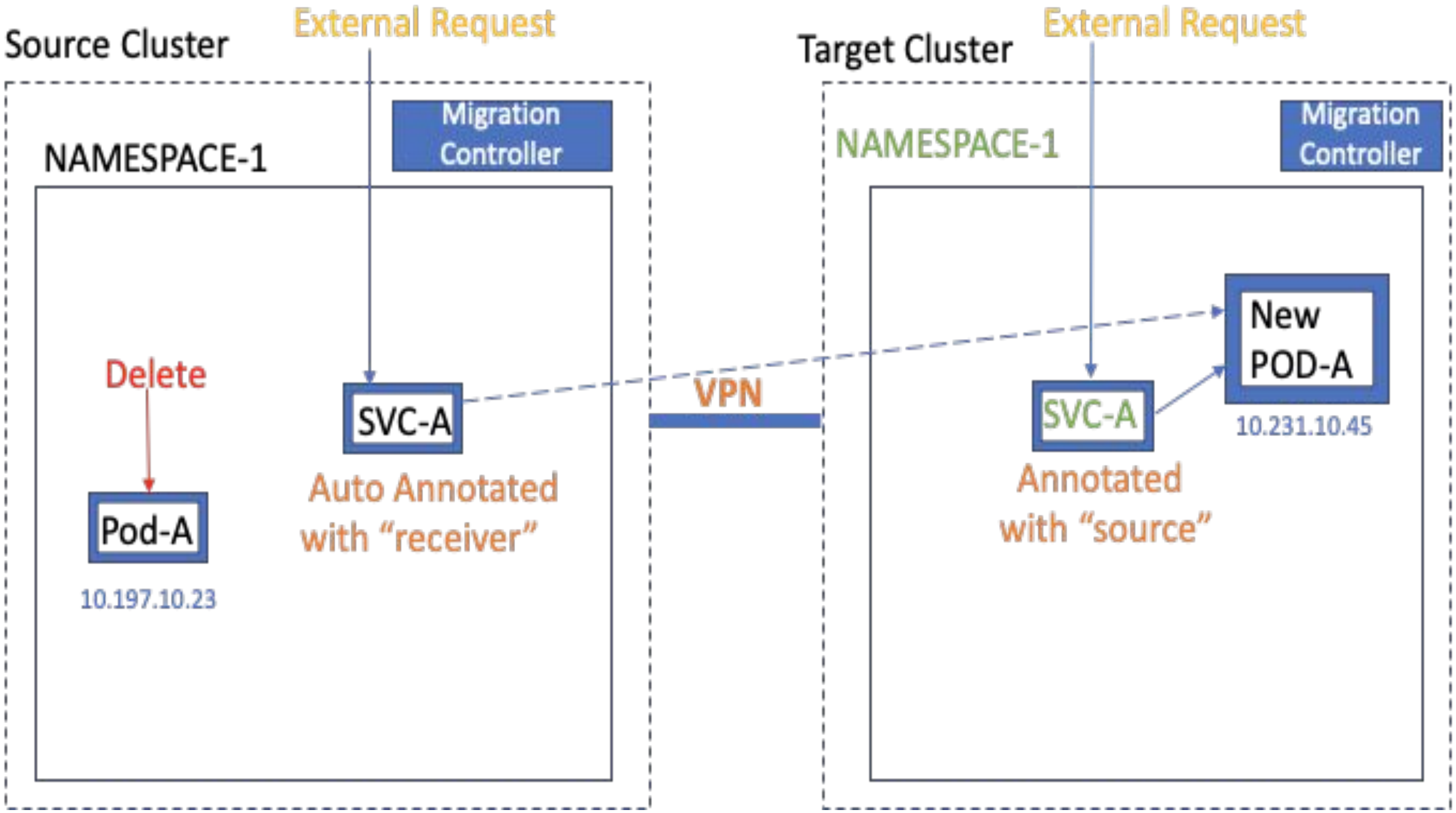

Source Cluster
Migration Controller
VPN
Target Cluster
Migration Controller

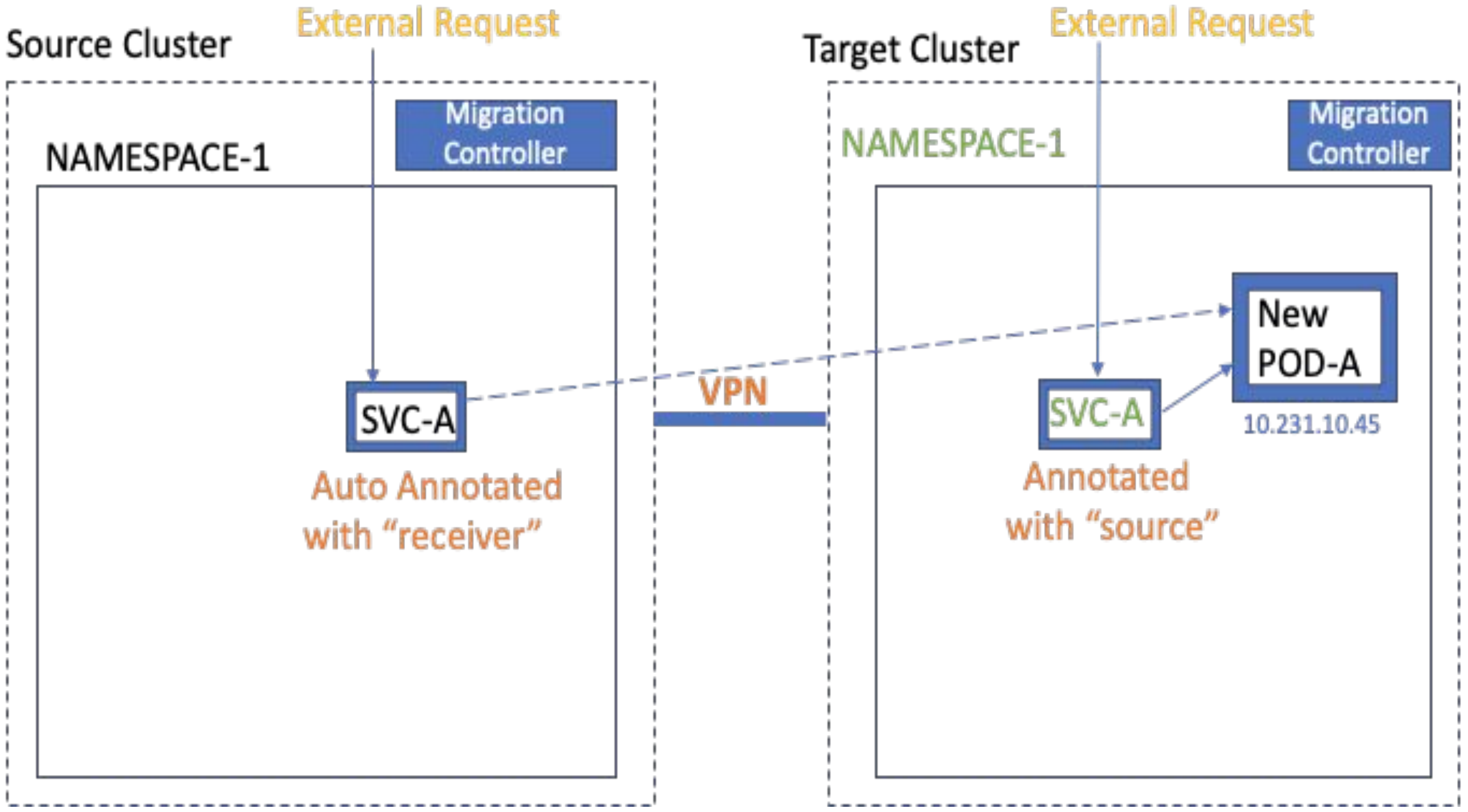

Source Cluster
NAMESPACE-1
Migration Controller
VPN
Target Cluster
NAMESPACE-1
Migration Controller
(Label: replicated: "true)


Source Cluster
Target Cluster
NAMESPACE-1
Migration Controller
NAMESPACE-1
Migration Controller
SVC-A
Pod-A
10.197.10.23
VPN
SVC-A
(Label : "replicated-true")


Source Cluster
External Request
NAMESPACE-1
Migration Controller
SVC-A
Pod-A
10.197.10.23
VPN
Target Cluster
External Request
NAMESPACE-1
Migration Controller
SVC-A
(Label : "replicated-true")


Source Cluster
External Request
Migration Controller
NAMESPACE-1
SVC-A
Pod-A
10.197.10.23
Annotate with "union"
VPN
Target Cluster
External Request
NAMESPACE-1
Migration Controller
New POD-A
10.231.10.45
SVC-A
Auto- Annotated with "union"


Source Cluster
External Request
Migration Controller
NAMESPACE-1
Delete
SVC-A
Pod-A
10.197.10.23
Auto Annotated with "receiver"
VPN
Target Cluster
External Request
NAMESPACE-1
Migration Controller
New POD-A
SVC-A
10.231.10.45
Annotated with "source"


Source Cluster
External Request
Migration Controller
NAMESPACE-1
SVC-A
Auto Annotated with “receiver”
VPN
Target Cluster
External Request
NAMESPACE-1
Migration Controller
New POD-A
SVC-A
10.231.10.45
Annotated with “source”

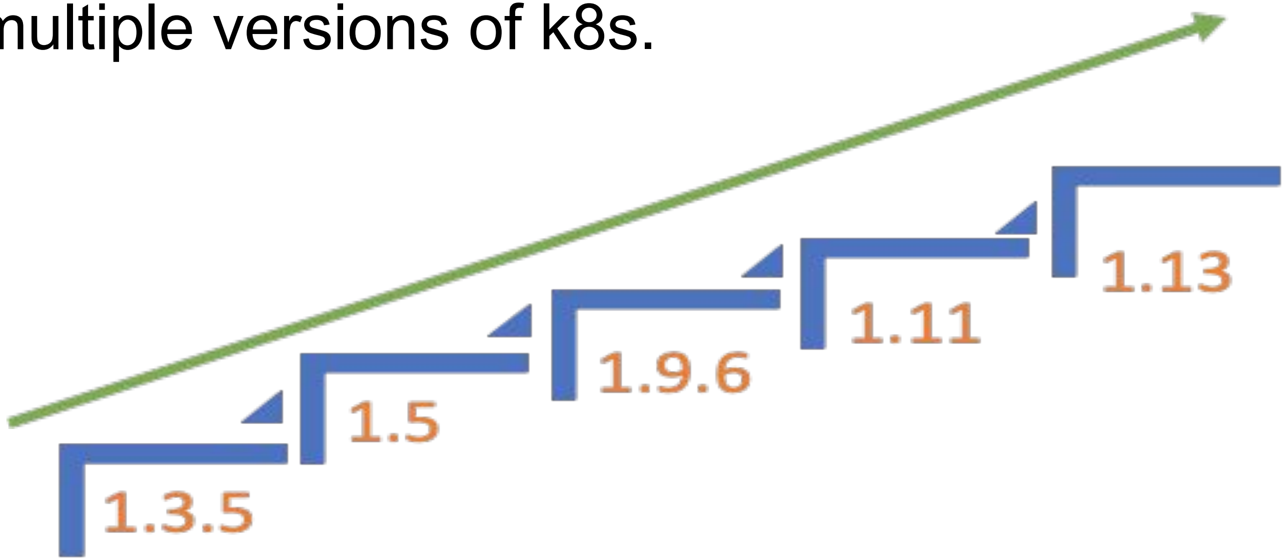
To not replicate the service use annotation:

“vmware.com/syndicate-mode: singular”

Connecting Kubernetes Cluster

1. Clusters should have different CIDR for pods
2. All the clusters should be connected with VPN, so that there is Pod to Pod connectivity across cluster
3. The Kubernetes API Server of every cluster should be reachable to other clusters.

We have now performed k8s upgrade of our cluster using migration controller for multiple versions of k8s.



Thank You.

If you have any further questions and/or want to discuss this, please reach us at ssinghvi@vmware.com or malhotrav@vmware.com

How to ruin an SRE-Dev relationship in 3 simple steps

Raushaniya Maksudova,
Site Reliability Engineer-Software Engineer, Google

SREcon Asia 2019

Agenda

- Step#0
- Step#1
- Step#2

Source: https://www.utest.com/status/71790

Step#0 - [Not] Understanding

Source: https://blog.hubspot.com/marketing/rant-content-strategists

Step#0 - [Not] Understanding

Source: https://blog.hubspot.com/marketing/rant-content-strategists

How to fix or prevent it? Practical tips

1) Write down SRE responsibilities and share with Devs.

2) Clearly communicate that SREs and Devs have the same end goal - users' happiness - but approach it differently.

3) Get involved in design development as early as possible.

How to fix or prevent it? Practical tips

1) Write down SRE responsibilities and share with Devs.

2) Clearly communicate that SREs and Devs have the same end goal - users' happiness - but approach it differently.

3) Get involved in design development as early as possible.

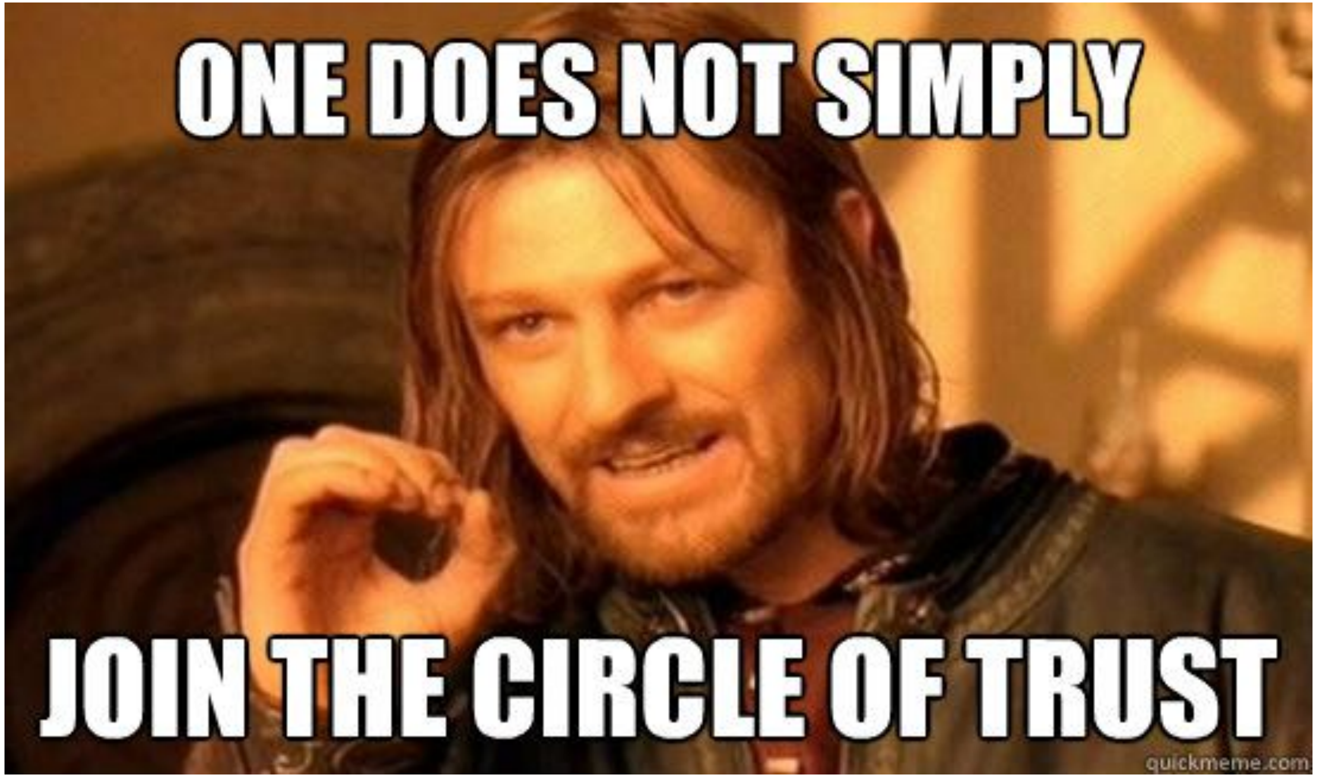
How to fix or prevent it? Practical tips

1) Write down SRE responsibilities and share with Devs.

2) Clearly communicate that SREs and Devs have the same end goal - users' happiness - but approach it differently.

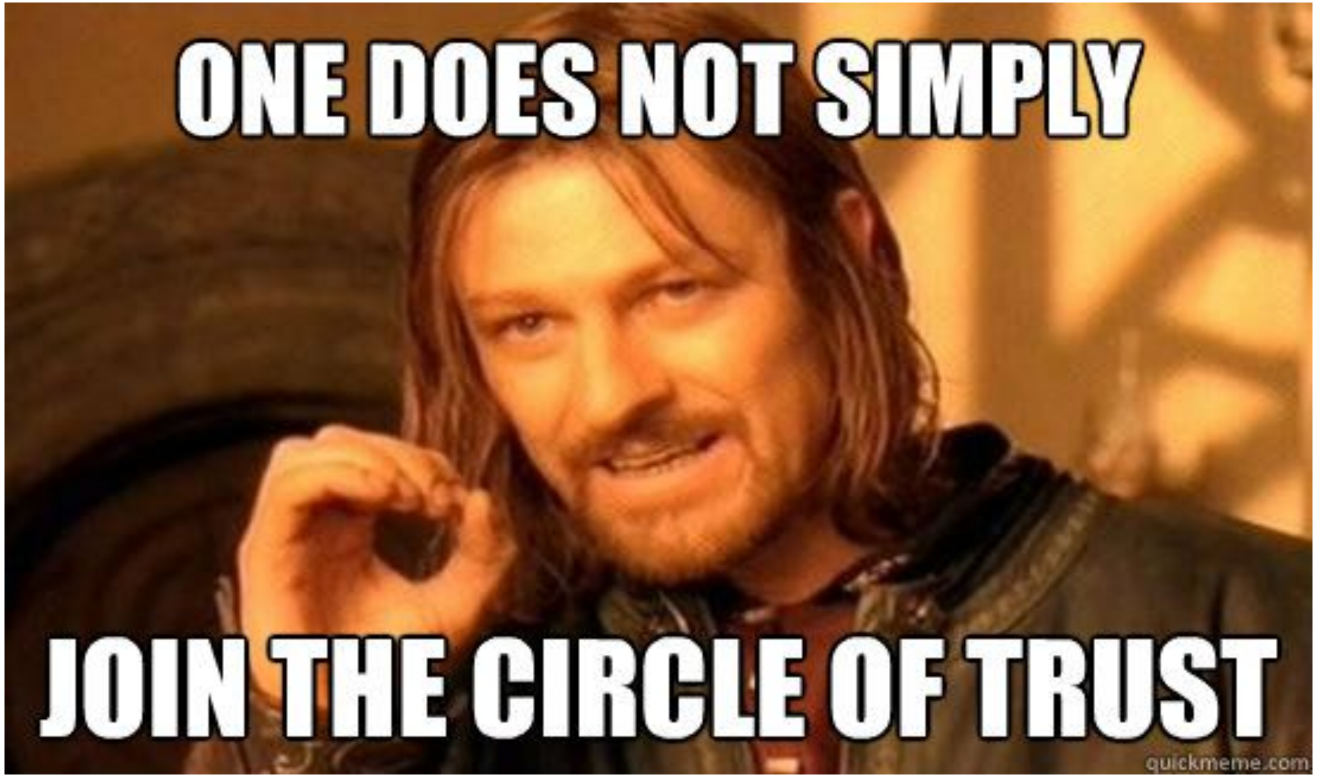
3) Get involved in design development as early as possible.

Step#1 - [Not] Trusting



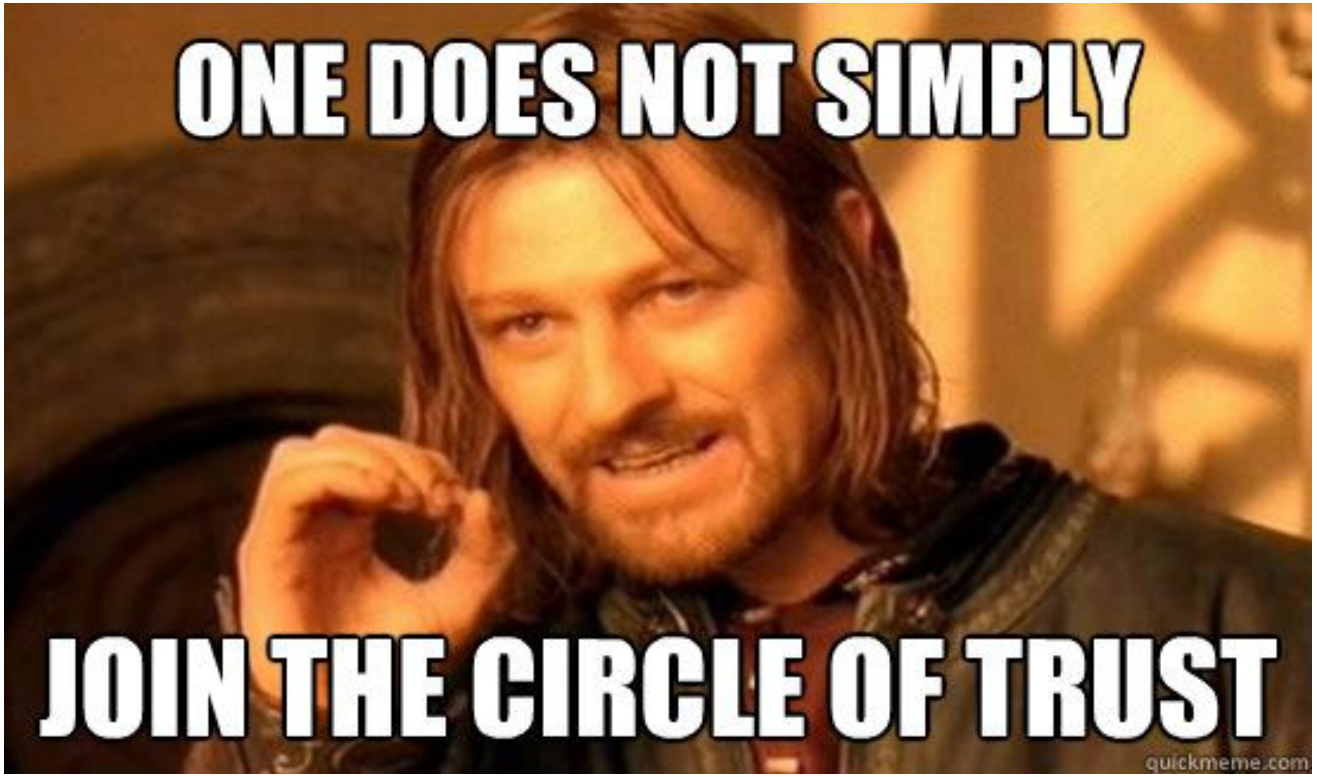
Source: http://www.quickmeme.com/meme/35zpzm

Step#1 - [Not] Trusting



Source: http://www.quickmeme.com/meme/35zpzm

Step#1 - [Not] Trusting



Source: http://www.quickmeme.com/meme/35zpzm

Google

How to fix or prevent it? Practical tips

1) Schedule a meeting to discuss all the pain points that Devs are dealing with at the moment.

2) Ease their production pain.

3) Put Devs oncall.

How to fix or prevent it? Practical tips

1) Schedule a meeting to discuss all the pain points that Devs are dealing with at the moment.

2) Ease their production pain.

3) Put Devs oncall.

How to fix or prevent it? Practical tips

1) Schedule a meeting to discuss all the pain points that Devs are dealing with at the moment.

2) Ease their production pain.

3) Put Devs oncall.

How to fix or prevent it? Practical tips

1) Schedule a meeting to discuss all the pain points that Devs are dealing with at the moment.

2) Ease their production pain.

3) Put Devs oncall.

Step#2 - [Not] Collaborating

Source: https://imgflip.com/i/2nxi0i

Step#2 - [Not] Collaborating

Source: https://imgflip.com/i/2nxi0i

How to fix or prevent it? Practical tips

1) Plan things you want to work on in the next quarter together with your Dev team.

2) Set up regular (bi-weekly/monthly) meetings with Dev team to stay updated, to answer each other's questions and offer help to each other. Over-communicate.

How to fix or prevent it? Practical tips

1) Plan things you want to work on in the next quarter together with your Dev team.

2) Set up regular (bi-weekly/monthly) meetings with Dev team to stay updated, to answer each other's questions and offer help to each other. Over-communicate.

How to fix or prevent it? Practical tips

1) Plan things you want to work on in the next quarter together with your Dev team.

2) Set up regular (bi-weekly/monthly) meetings with Dev team to stay updated, to answer each other’s questions and offer help to each other. Over-communicate.

Revision:

- Understanding
- Trusting
- Collaborating

An Effective Agile SRE Workflow

Jay Chin
Engineering Lead - SRE
jay.chin@grab.com *@jaychin*

What does an SRE do ?

1. Product
2. Development
3. Capacity Planning
4. Testing + Release Procedures
5. Postmortem / RCA
6. Incident Response
7. Monitoring
8. Support Requests
9. Code Reviews
10. Etc

What does an SRE do ?

1. Product
2. Development
3. Capacity Planning
4. Testing + Release Procedures
5. Postmortem / RCA
6. Incident Response
7. Monitoring
8. Support Requests
9. Code Reviews
10. Etc

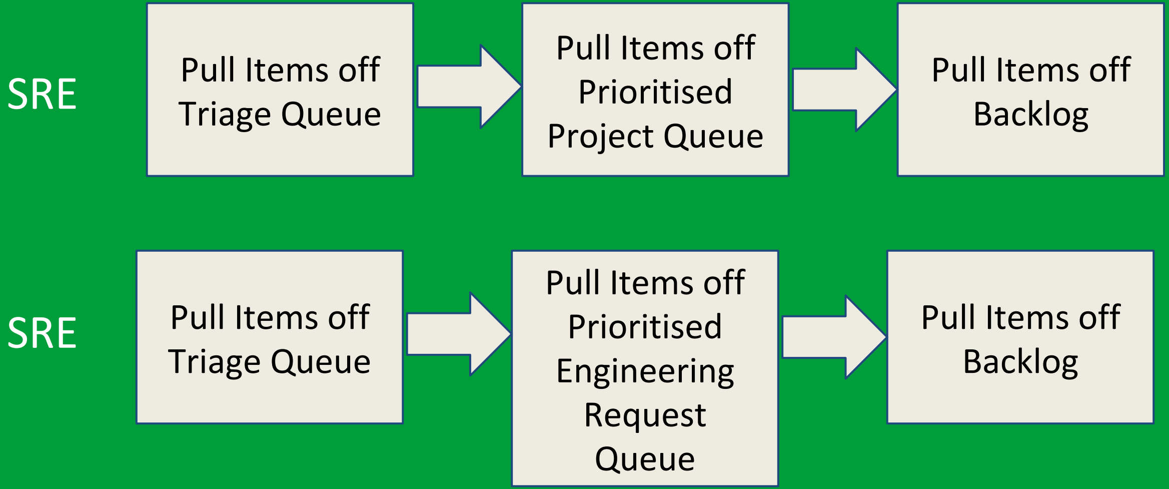
Scrum ?

"Scrum" (CC BY 2.0) by Conor Lawless

Kanban ? Scrumban ?

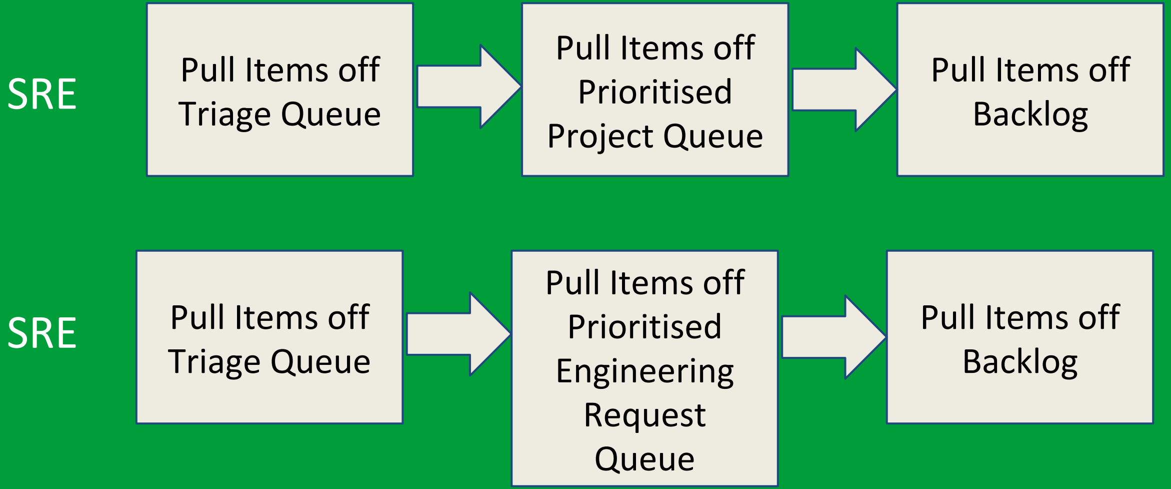
Multiple Boards for Different Workflows

1. SRE Main Project Board
2. Engineering Request Board
3. Triage Queue


SRE
Pull Items off Triage Queue
Pull Items off Prioritised Project Queue
Pull Items off Backlog
SRE
Pull Items off Triage Queue
Pull Items off Prioritised Engineering Request Queue
Pull Items off Backlog

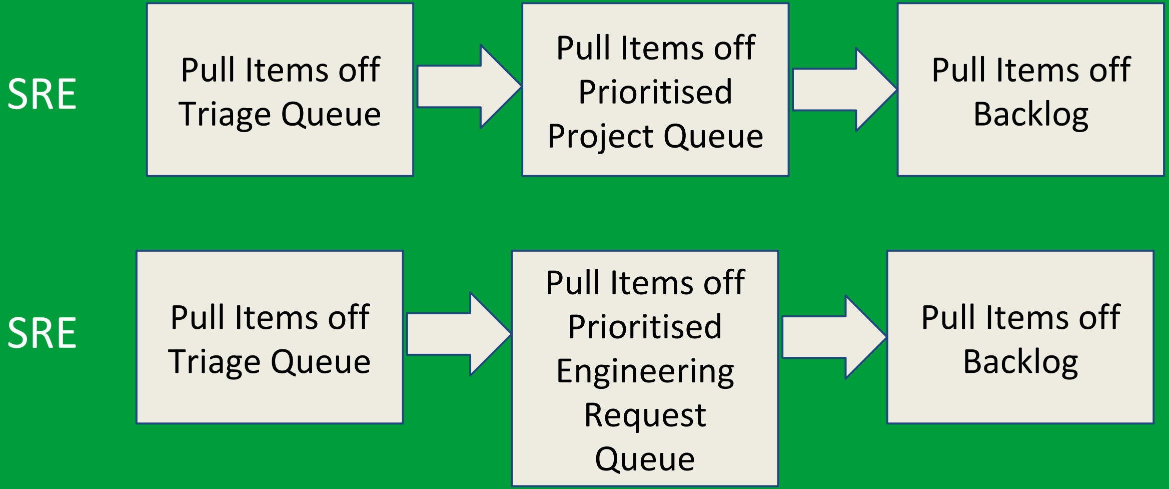
Multiple Boards for Different Workflows

1. SRE Main Project Board
2. Engineering Request Board
3. Triage Queue


SRE
Pull Items off Triage Queue
Pull Items off Prioritised Project Queue
Pull Items off Backlog
SRE
Pull Items off Triage Queue
Pull Items off Prioritised Engineering Request Queue
Pull Items off Backlog

Multiple Boards for Different Workflows

1. SRE Main Project Board
2. Engineering Request Board
3. Triage Queue


SRE
Pull Items off Triage Queue
Pull Items off Prioritised Project Queue
Pull Items off Backlog
SRE
Pull Items off Triage Queue
Pull Items off Prioritised Engineering Request Queue
Pull Items off Backlog

SRE Main Projects

- This is where SRE projects go (Automation, Upgrades, Product Spikes)
- Breakdown task into small chunks (e.g. 2 days). Stuck/Complex tasks quickly become apparent in your daily standups.

Pro Tip : Limit Work In Progress (WIP) !

SRE Main Projects

- This is where SRE projects go (Automation, Upgrades, Product Spikes)
- Breakdown task into small chunks (e.g. 2 days). Stuck/Complex tasks quickly become apparent in your daily standups.

Pro Tip : Limit Work In Progress (WIP) !

Engineering Requests (Toil Board)

- Toil Work
- Categorise and Quantify Everything !
- Goal is to remove this board altogether.

Engineering Requests (Toil Board)

- Toil Work
- Categorise and Quantify Everything !
- Goal is to remove this board altogether.

Engineering Requests (Toil Board)

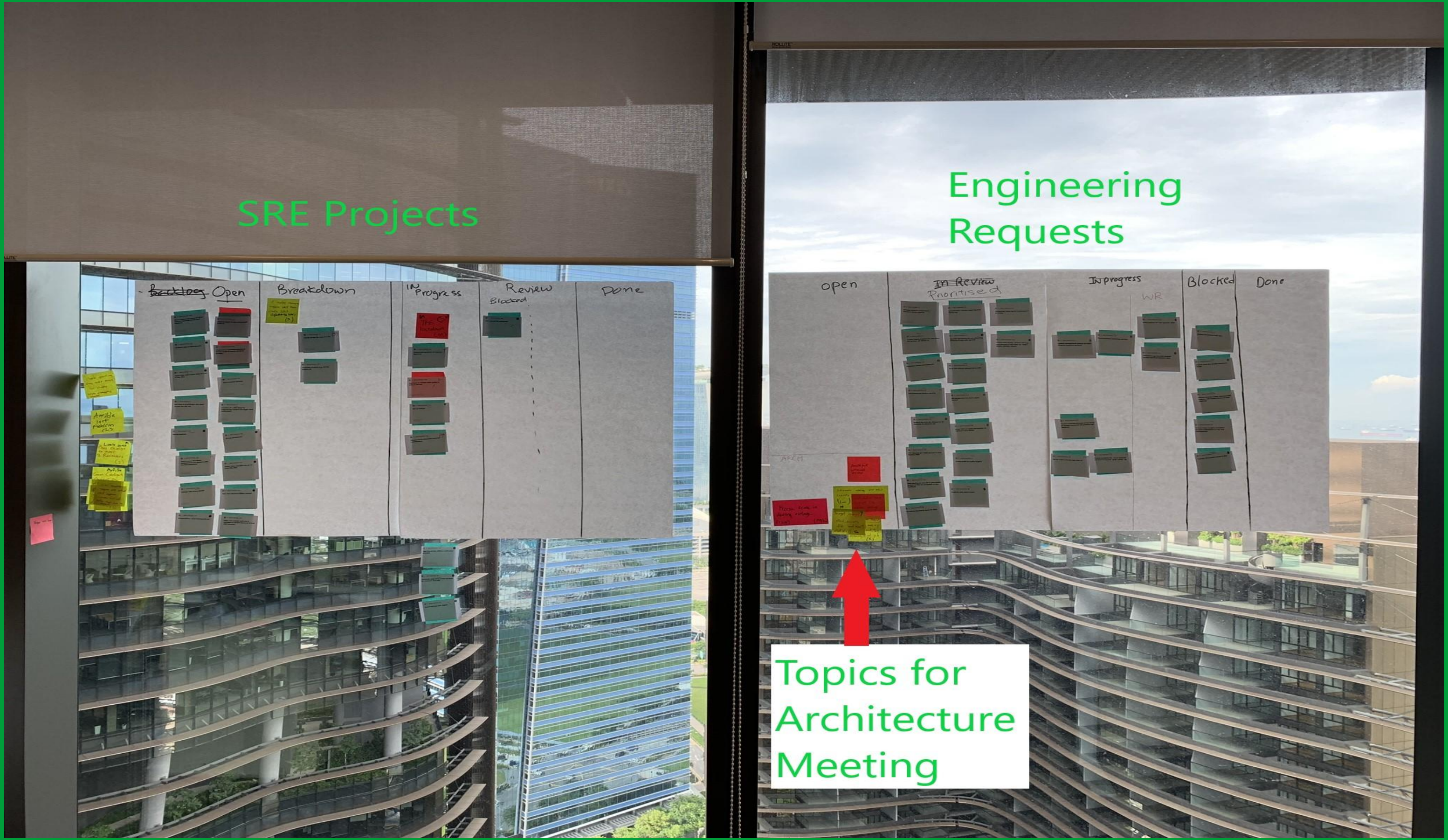
- Toil Work
- Categorise and Quantify Everything !
- Goal is to remove this board altogether.

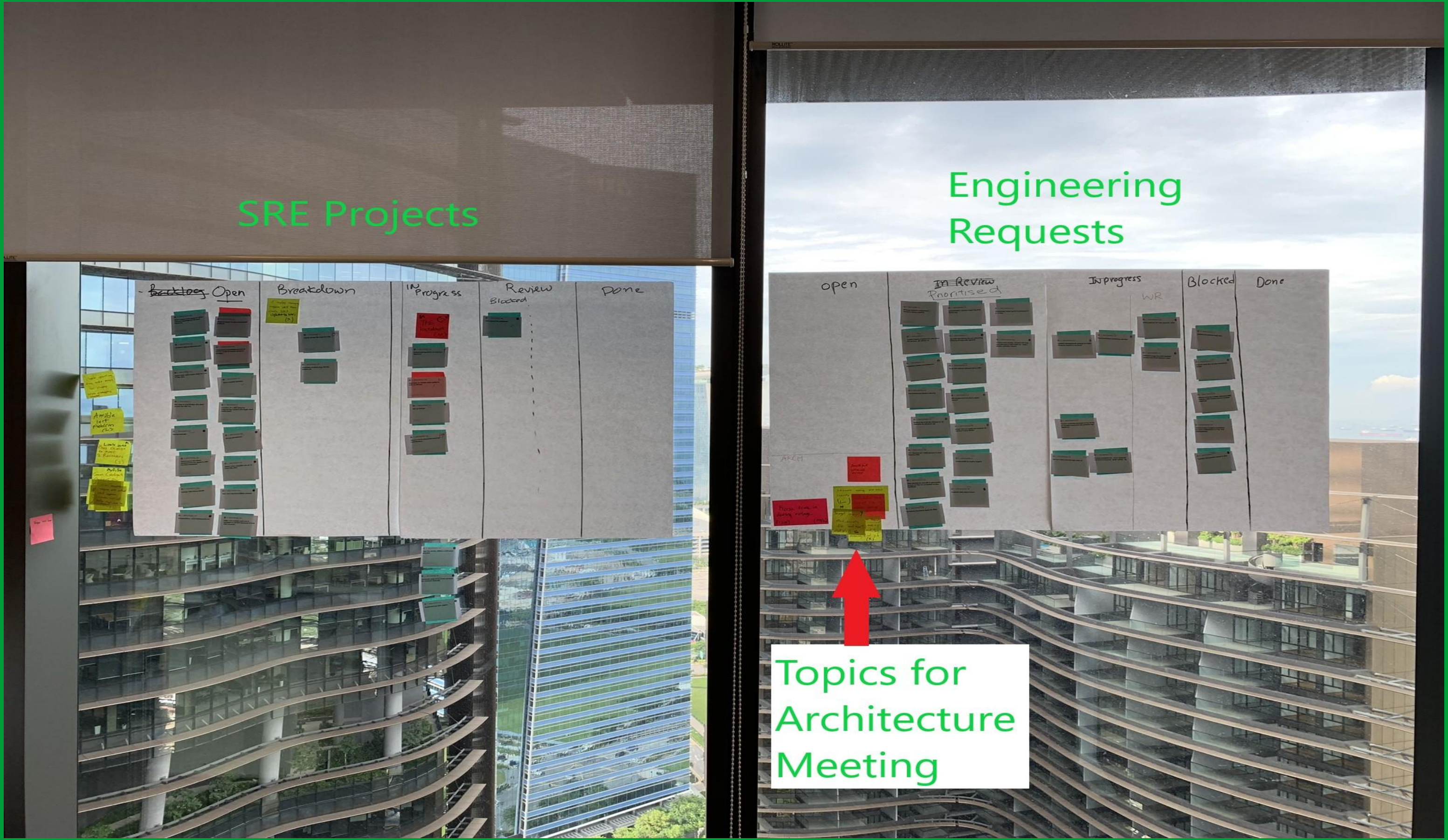
Use an Interrupt Shield

1. Shield from ad-hoc requests and user support communication (online and walk-ups)
2. This permits other team members to focus on projects/automation/etc

Use an Interrupt Shield

1. Shield from ad-hoc requests and user support communication (online and walk-ups)
2. This permits other team members to focus on projects/automation/etc


SRE Projects
~~Backlog~~ Open
Breakdown
In Progress
Review
Blocked
Done
Engineering Requests
open
~~In Review~~
Prioritised
In progress
WR
Blocked
Done
ARCH
Topics for Architecture Meeting


SRE Projects
~~Backlog~~ Open
Breakdown
In Progress
Review
Blocked
Done
Engineering Requests
Open
~~In Review~~ Prioritised
In progress
WR
Blocked
Done
ARCH
Topics for Architecture Meeting

Final Tips

- No one-size-fits-all solution: observe, measure, and adapt.
- Adapt Kanban columns to your workflow as you go along.
- JIRA has a print feature to use on your physical board.

Final Tips

- Online boards (JIRA or Trello) might be more suitable for remote teams.
- Set-up policies/conditions for a sticky to enter or leave a column.
- Limit number of stickies in other columns too (not just in-progress).

Thank You

jay.chin@grab.com

We are Hiring !

UBISOFT

5 ACTIONS FOR TRAINING YOUR SRE TEAM

SRECON19 ASIA

DORIAN BASUYAU
IT ONLINE MANAGER,
UBISOFT IT
UBISOFT SINGAPORE

@dorian_bas

UBISOFT

5 ACTIONS FOR TRAINING YOUR SRE TEAM

SRECON19 ASIA

DORIAN BASUYAU
IT ONLINE MANAGER,
UBISOFT IT
UBISOFT SINGAPORE

@dorian_bas

KEEP UP LEARNING !

YOUR SRE TEAM MUST LEARN
TO STAY EFFICIENT

KEEP UP LEARNING !

YOUR SRE TEAM MUST LEARN
TO STAY EFFICIENT

ACTION 1

HAVE A SAFE PLACE TO LEARN

ACTION 1

HAVE A SAFE PLACE TO LEARN

ACTION 2

KNOW YOUR SRE’S

ACTION 2

KNOW YOUR SRE’S

DISSOCIATE IT FROM ANY PERFORMANCE EVALUATIONS

(SERIOUSLY)

DISSOCIATE IT FROM ANY PERFORMANCE EVALUATIONS

(SERIOUSLY)

ACTION 3

SET YOUR TECHNOLOGY FOCUS

CHOOSE WISELY

TRAINING IS SUPPORTING YOUR STRATEGY.

CHOOSE WISELY

TRAINING IS SUPPORTING YOUR STRATEGY.

ACTION 4

CREATE TRAINING WORKGROUPS

TRAINING WORKGROUPS TO KEEP YOUR SRE’S ENGAGED

LEARNING ALONE IS NOT ALWAYS FUN

TRAINING WORKGROUPS TO KEEP YOUR SRE'S ENGAGED

LEARNING ALONE IS NOT ALWAYS FUN

ACTION 5

TRACK THE PROGRESS

SHOW THE PROGRESS TO YOUR TEAM

EVALUATE THE LEARNING PROGRESSION ON REGULAR BASIS (I.E EVERY 6 MONTHS).

SHOW THE PROGRESS TO YOUR TEAM

EVALUATE THE LEARNING PROGRESSION ON REGULAR BASIS (I.E EVERY 6 MONTHS).

THANK YOU

UBISOFT

@dorian_bas

stackoverflow

Managing Terraform State

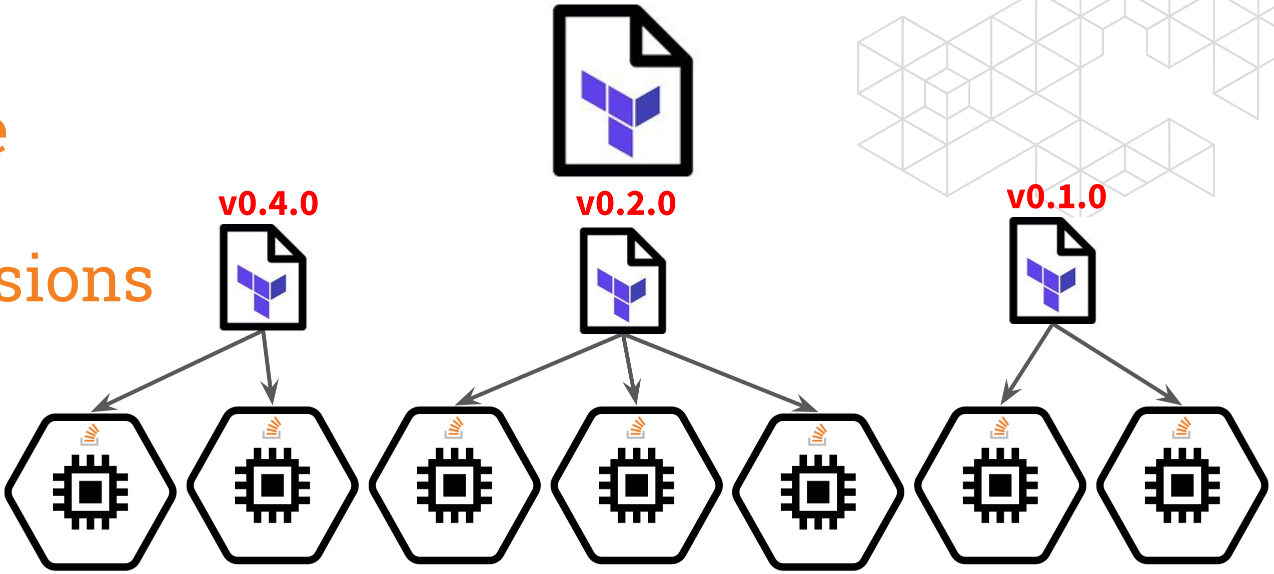
Mark Henderson - Stack Overflow

@thefarseeker

Terraform Module

With multiple versions

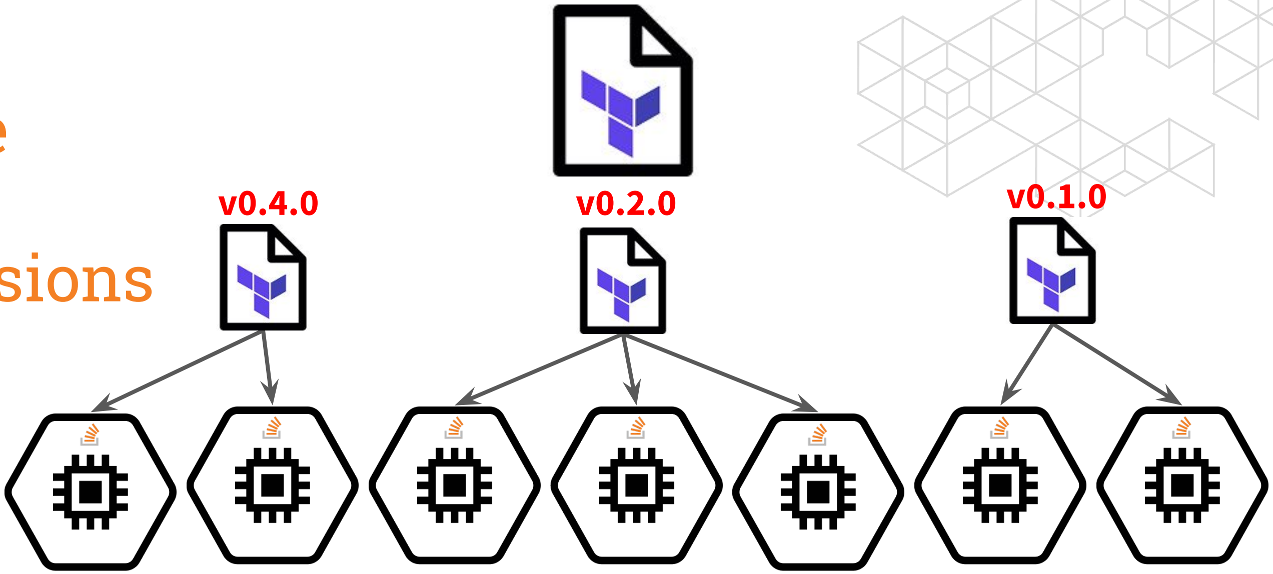
Deployed multiple times



Terraform Module

With multiple versions

Deployed multiple times

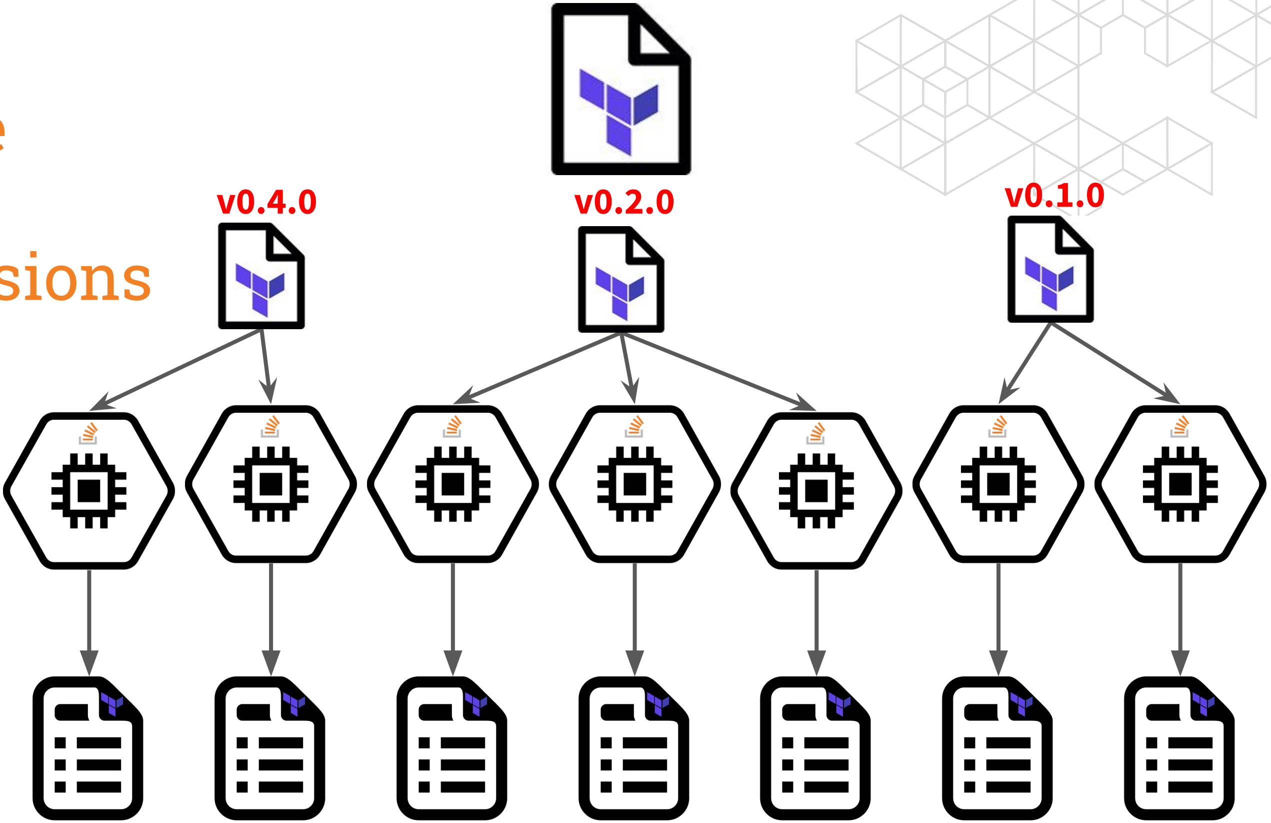


Terraform Module

With multiple versions

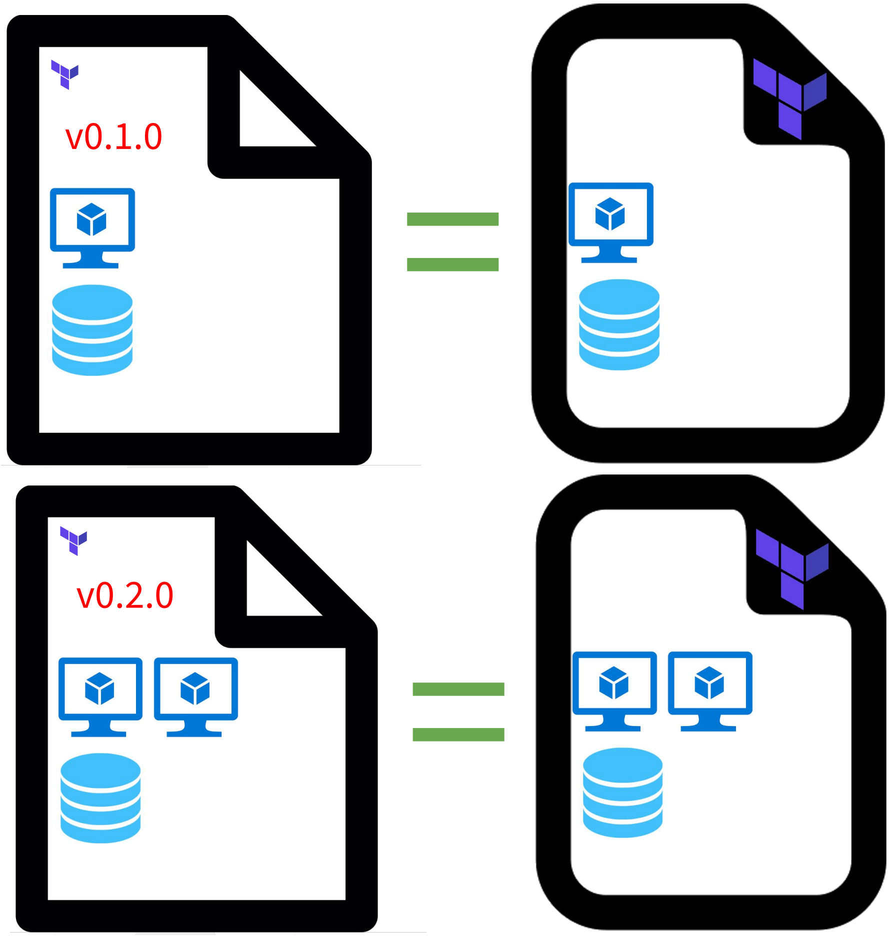
Deployed
multiple times

Each with their
own state



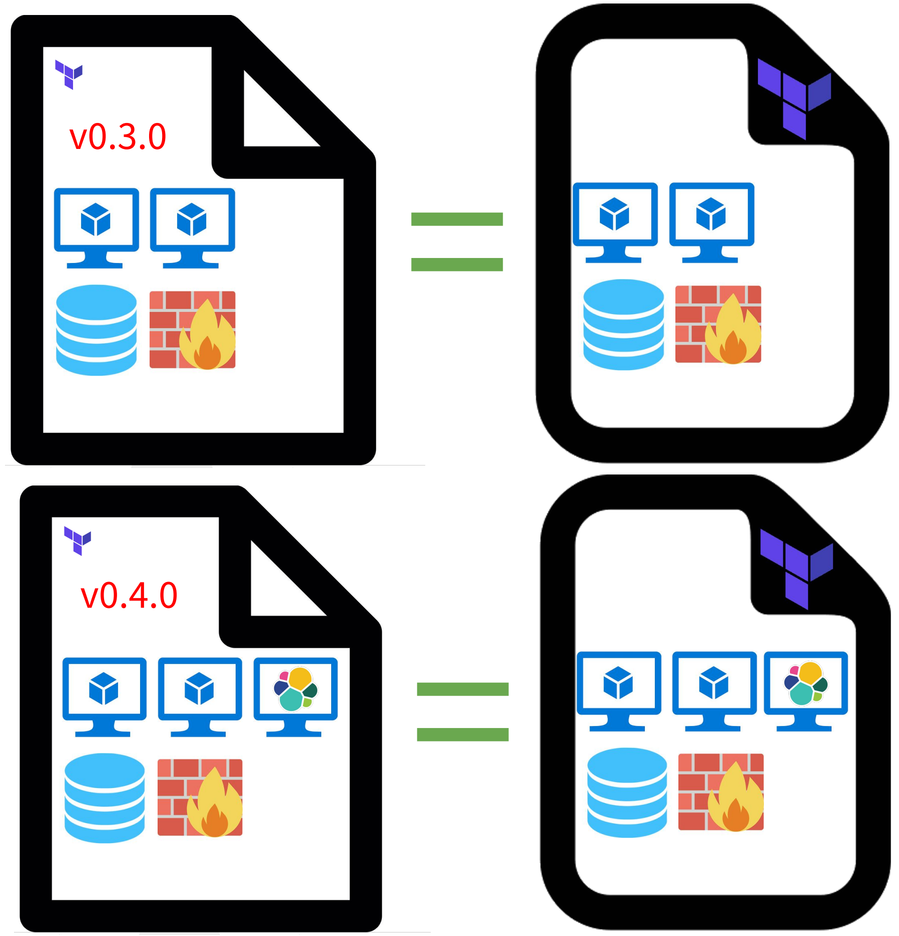
Terraform Module

State File



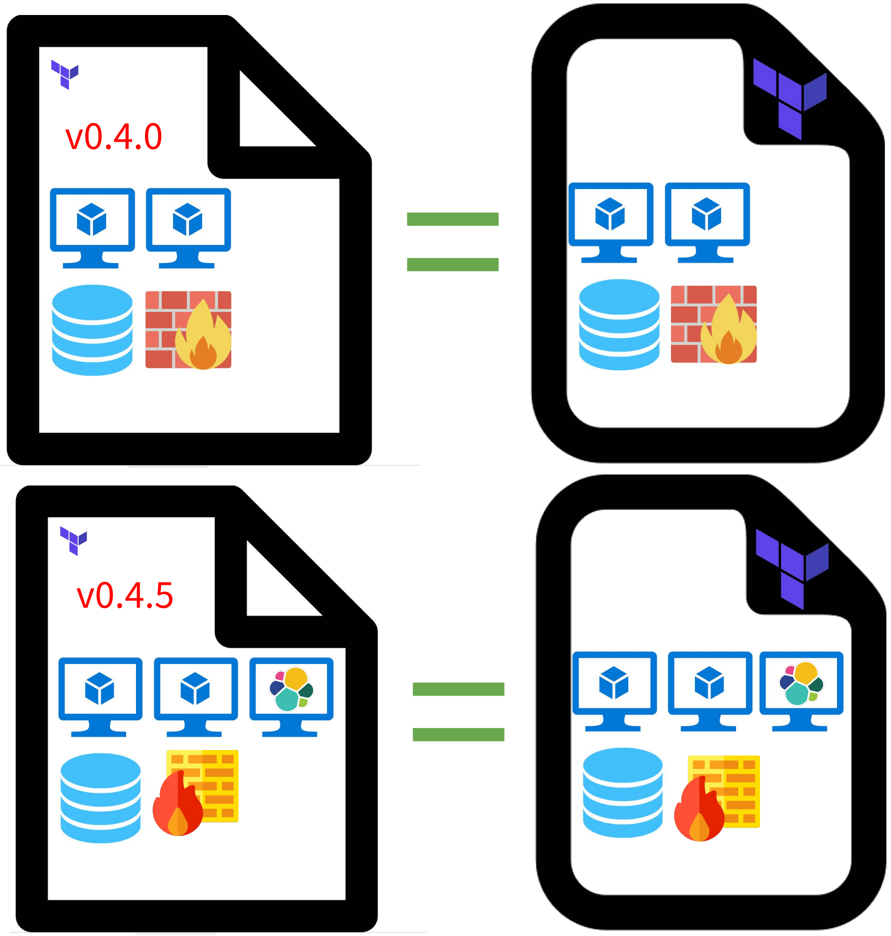
Terraform Module

State File



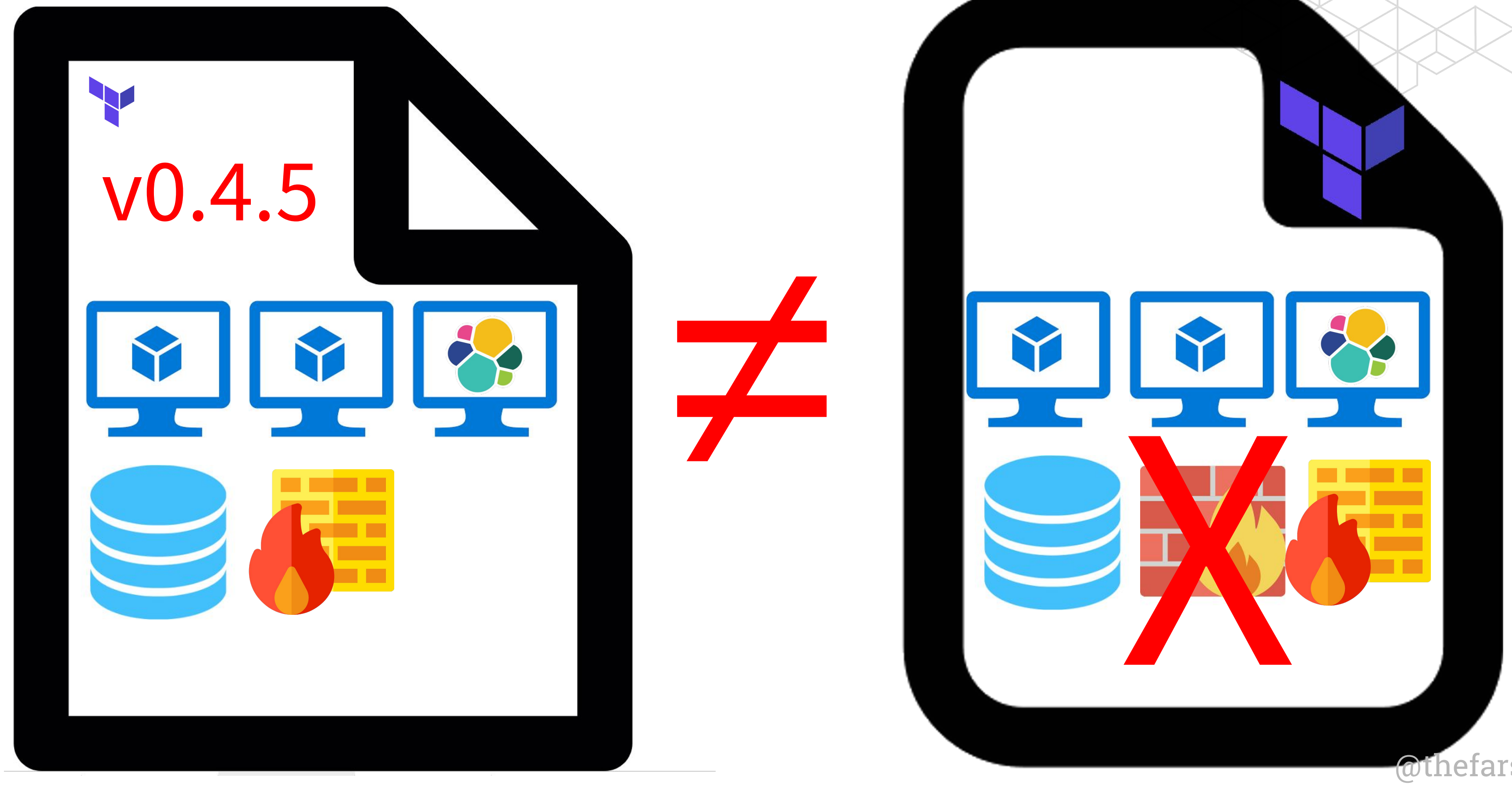
Terraform Module

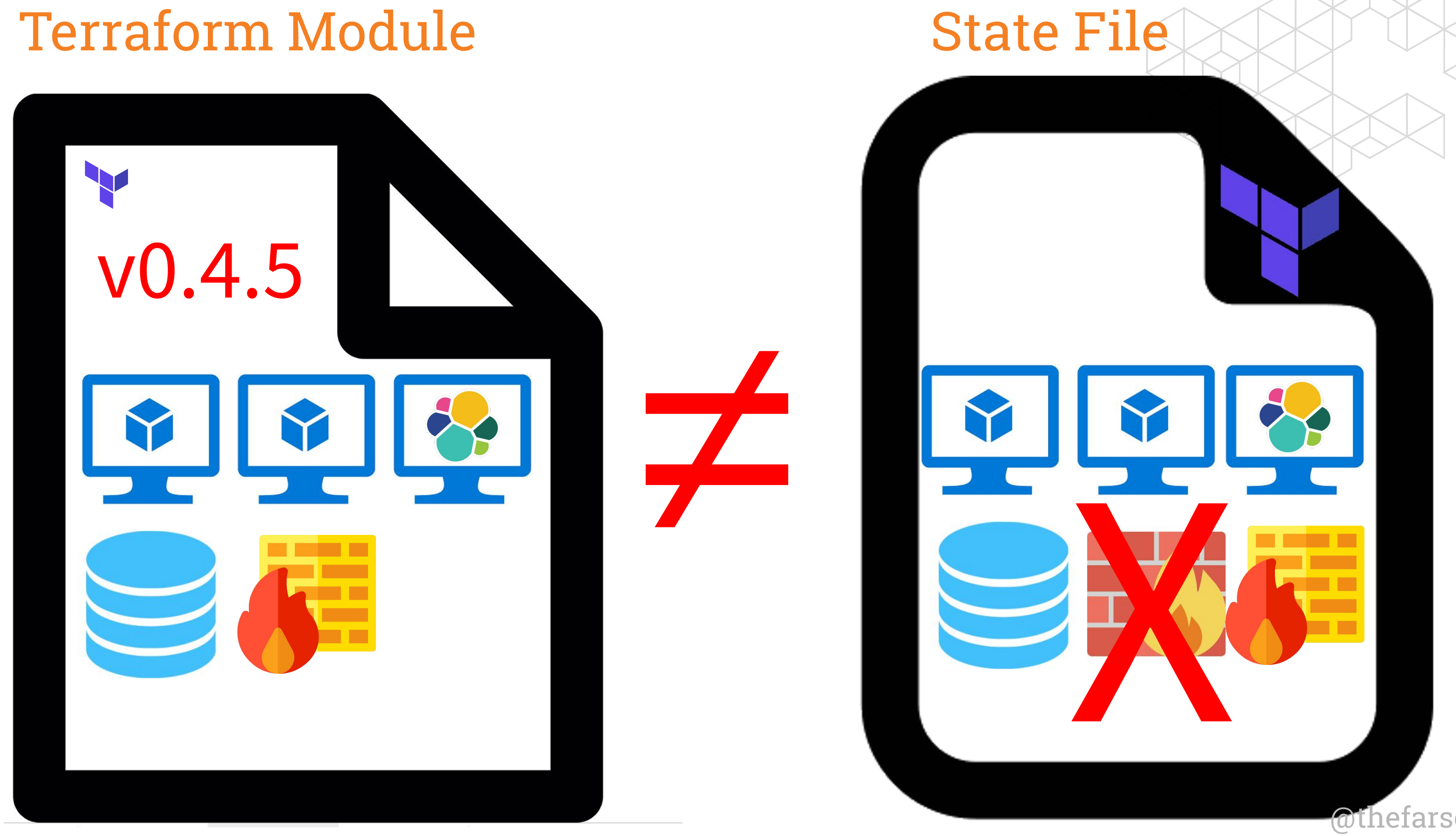
State File



Terraform Module

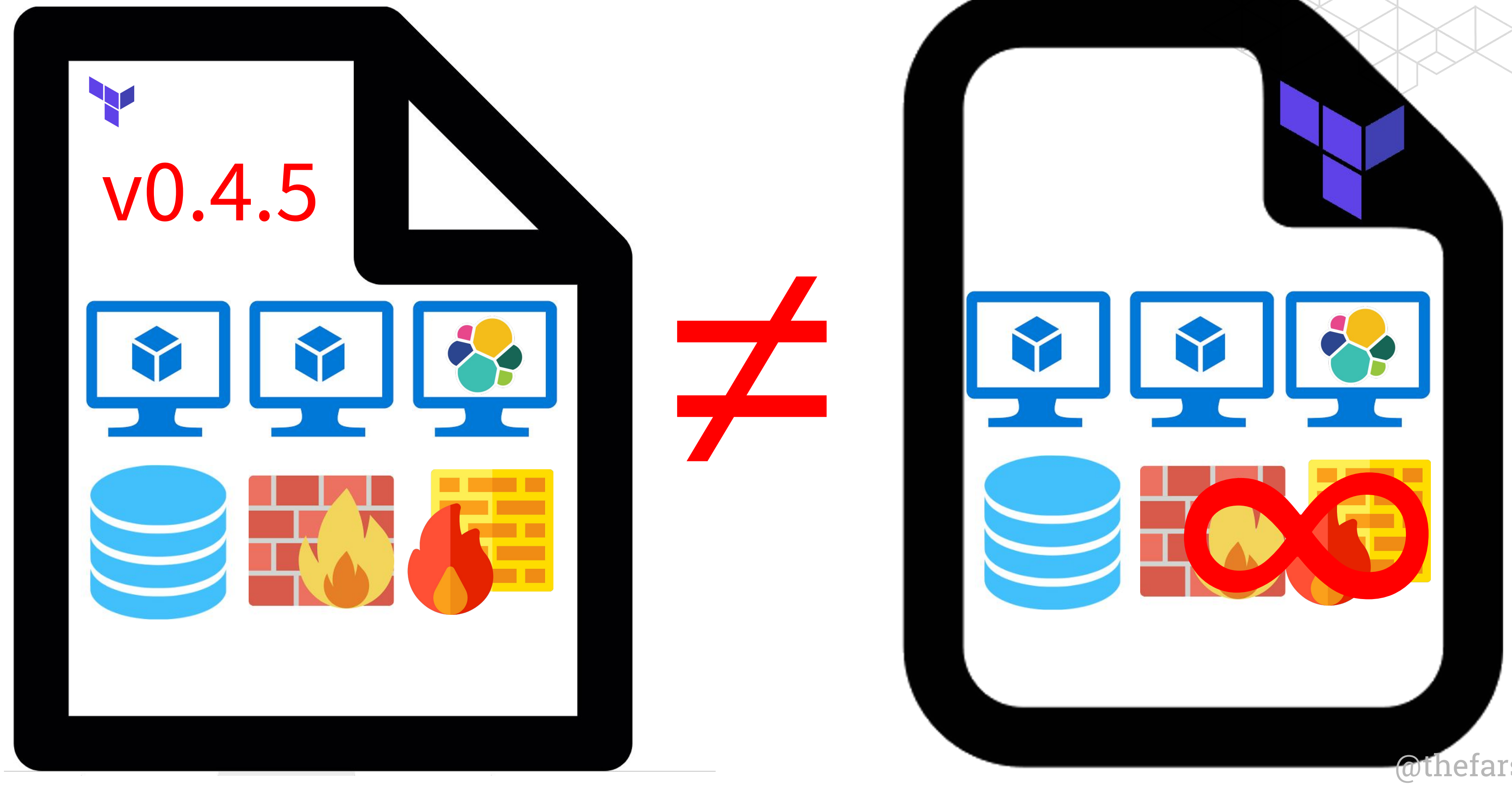
State File




Terraform Module
v0.4.5
≠
State File

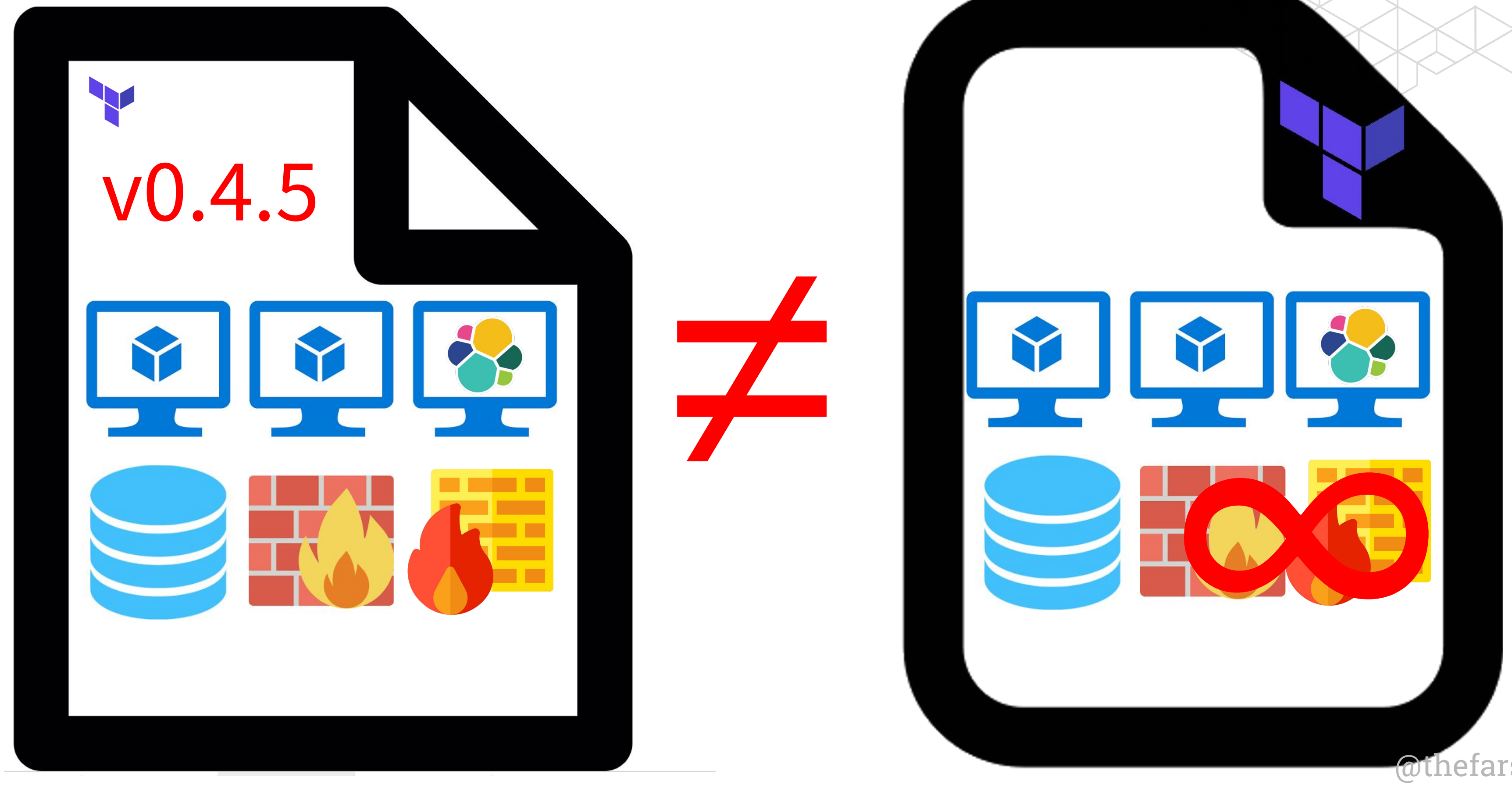
Terraform Module

State File

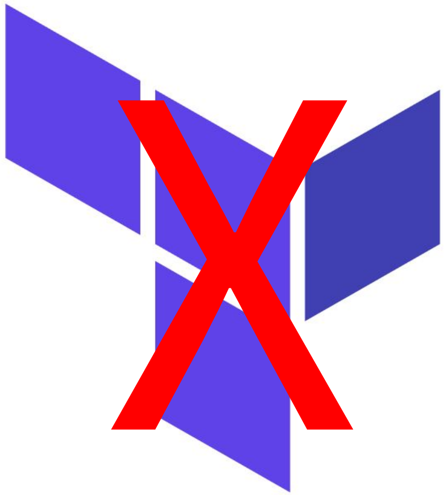


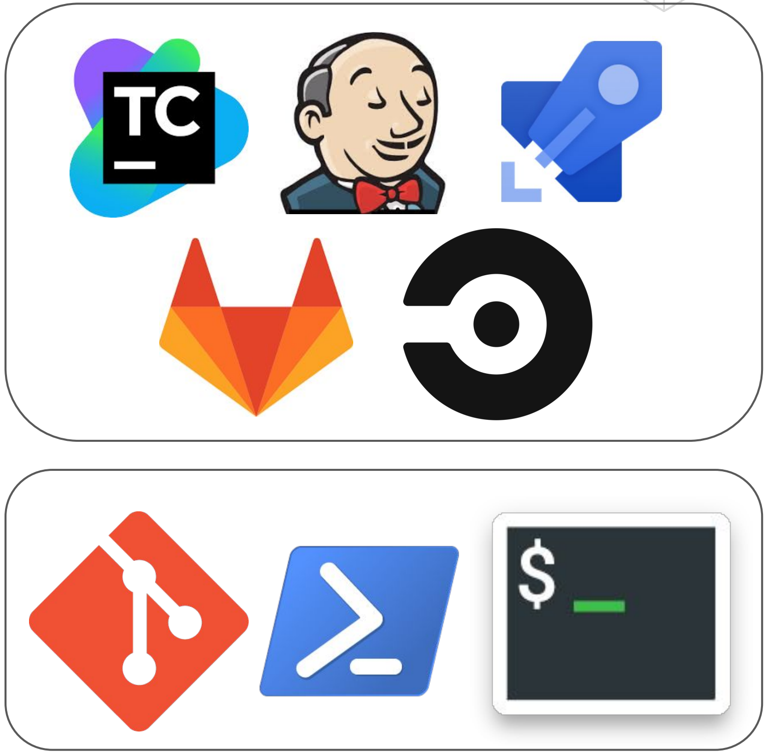
Terraform Module

State File

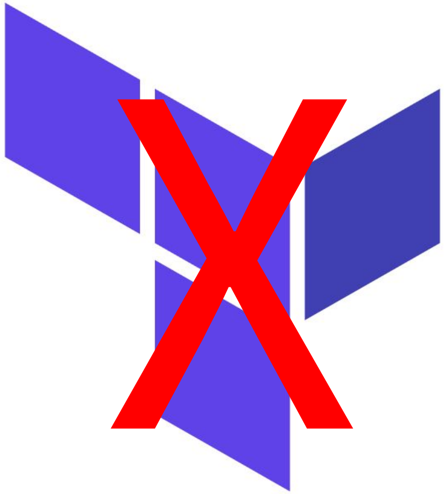


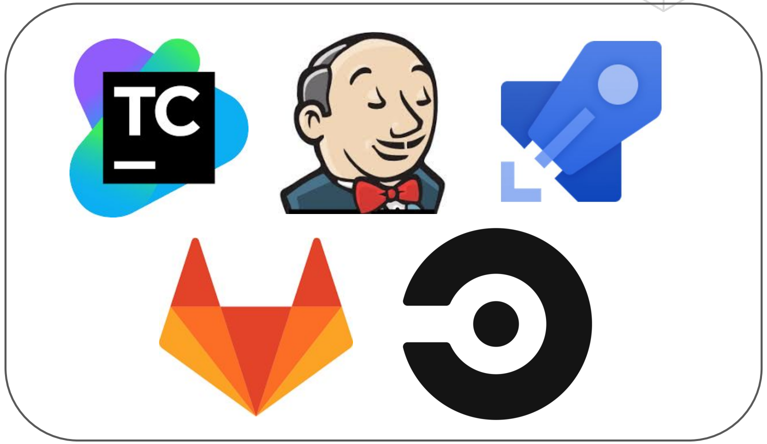
Migration support



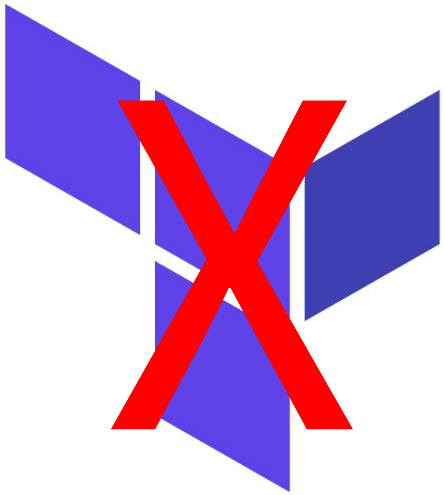


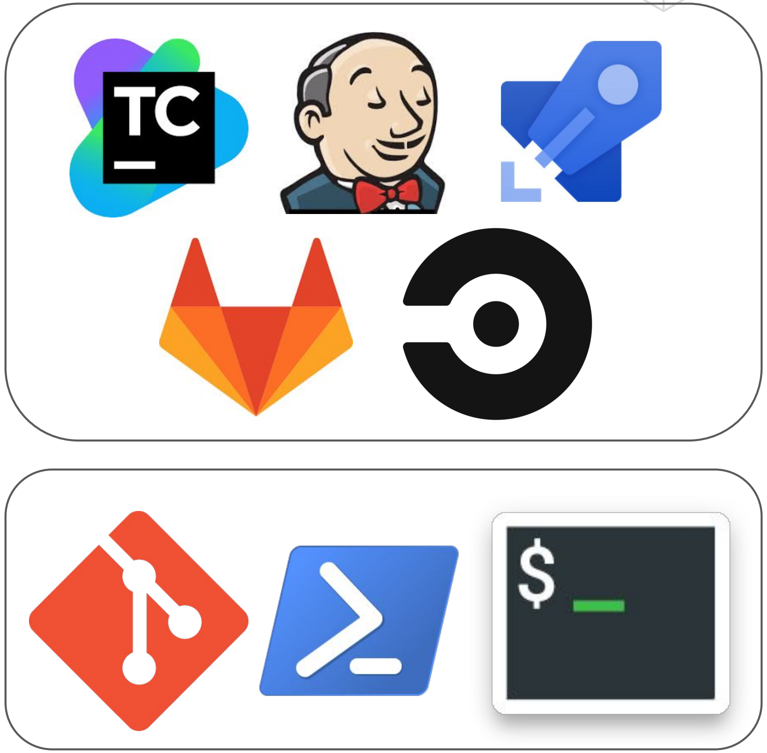
Migration support



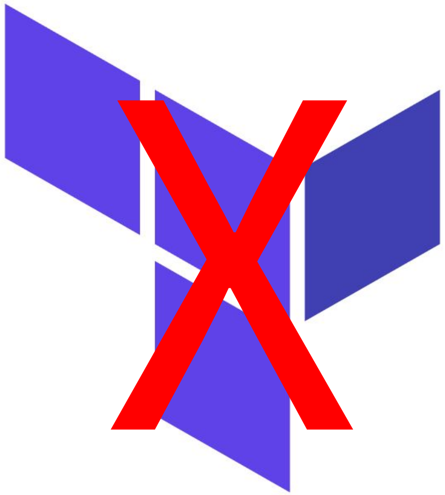


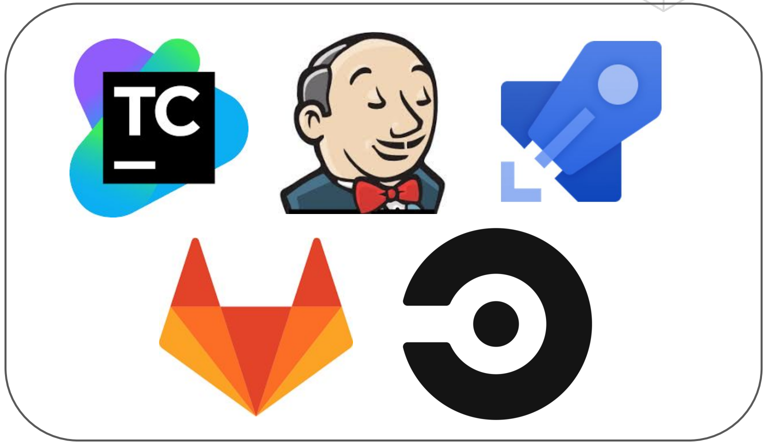
Migration support



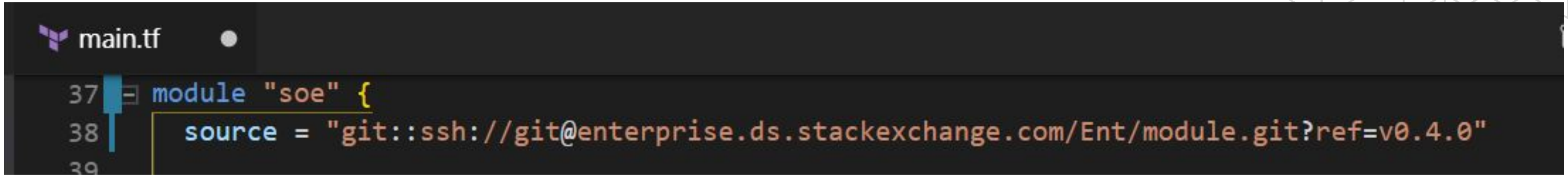


Migration support



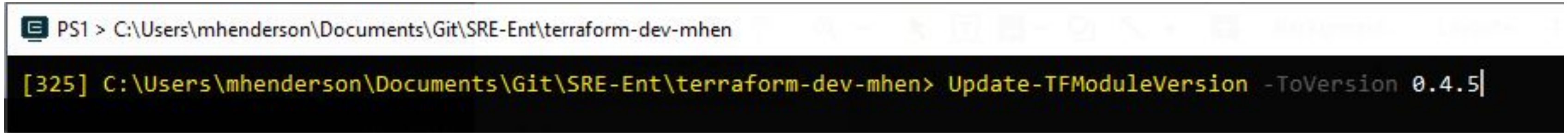


Example Migration



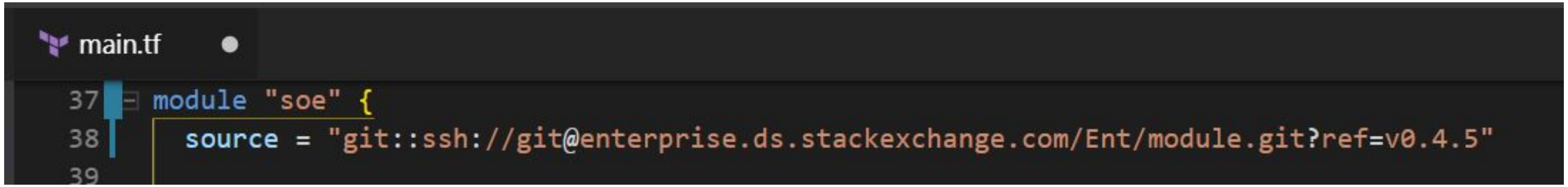
main.tf ●

```
module "soe" {
  source = "git::ssh://git@enterprise.ds.stackexchange.com/Ent/module.git?ref=v0.4.0"
```



PS1 > C:\Users\mhenderson\Documents\Git\SRE-Ent\terraform-dev-mhen

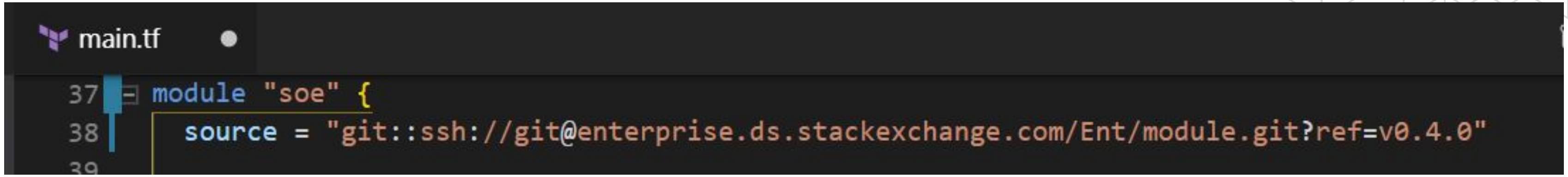
```
[325] C:\Users\mhenderson\Documents\Git\SRE-Ent\terraform-dev-mhen> Update-TFModuleVersion -ToVersion 0.4.5
```



main.tf ●

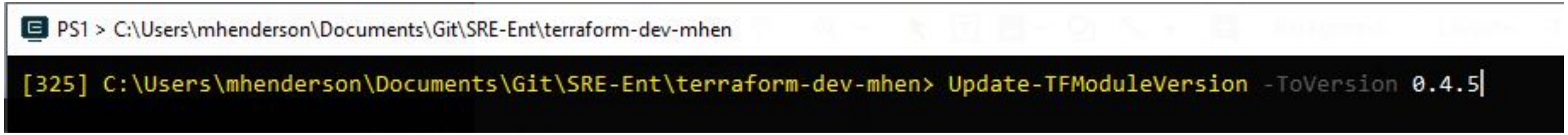
```
module "soe" {
  source = "git::ssh://git@enterprise.ds.stackexchange.com/Ent/module.git?ref=v0.4.5"
```

Example Migration



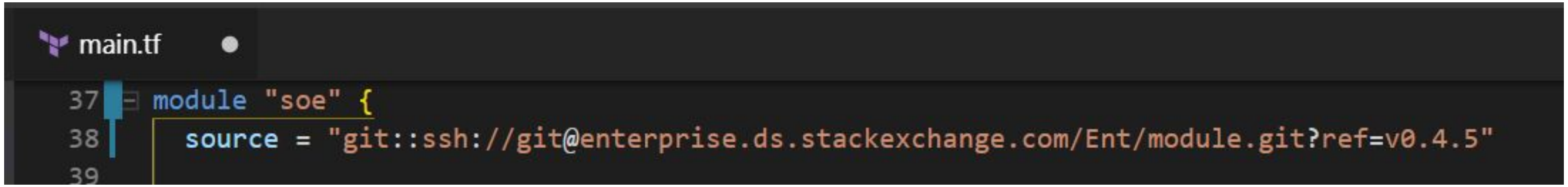
main.tf ●

```
module "soe" {
  source = "git::ssh://git@enterprise.ds.stackexchange.com/Ent/module.git?ref=v0.4.0"
```



PS1 > C:\Users\mhenderson\Documents\Git\SRE-Ent\terraform-dev-mhen

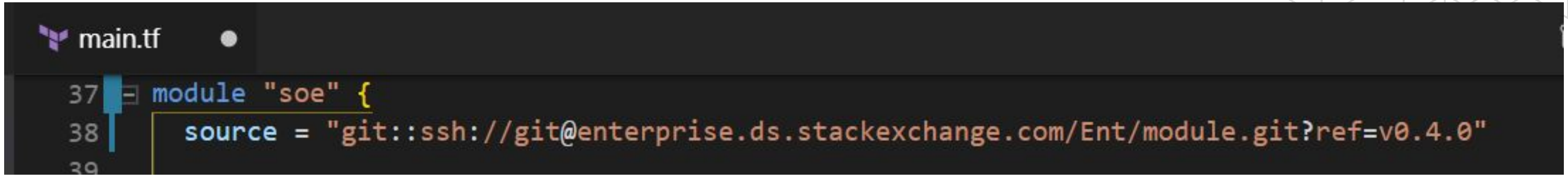
```
[325] C:\Users\mhenderson\Documents\Git\SRE-Ent\terraform-dev-mhen> Update-TFModuleVersion -ToVersion 0.4.5
```



main.tf ●

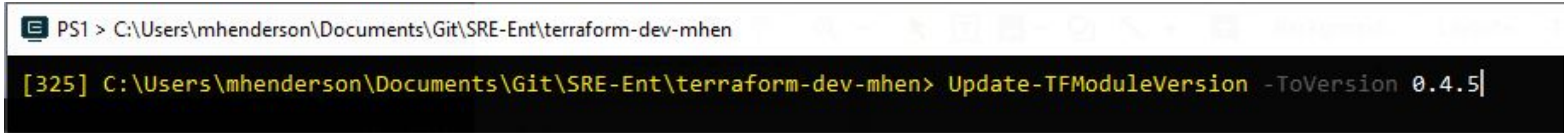
```
module "soe" {
  source = "git::ssh://git@enterprise.ds.stackexchange.com/Ent/module.git?ref=v0.4.5"
```

Example Migration



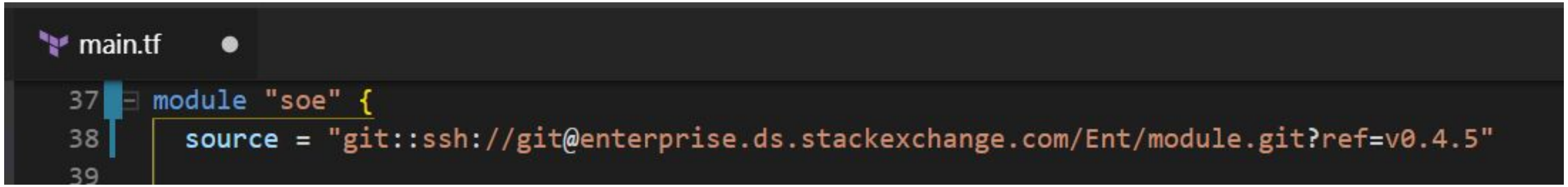
main.tf

```
module "soe" {
  source = "git::ssh://git@enterprise.ds.stackexchange.com/Ent/module.git?ref=v0.4.0"
```



PS1 > C:\Users\mhenderson\Documents\Git\SRE-Ent\terraform-dev-mhen

```
[325] C:\Users\mhenderson\Documents\Git\SRE-Ent\terraform-dev-mhen> Update-TFModuleVersion -ToVersion 0.4.5
```

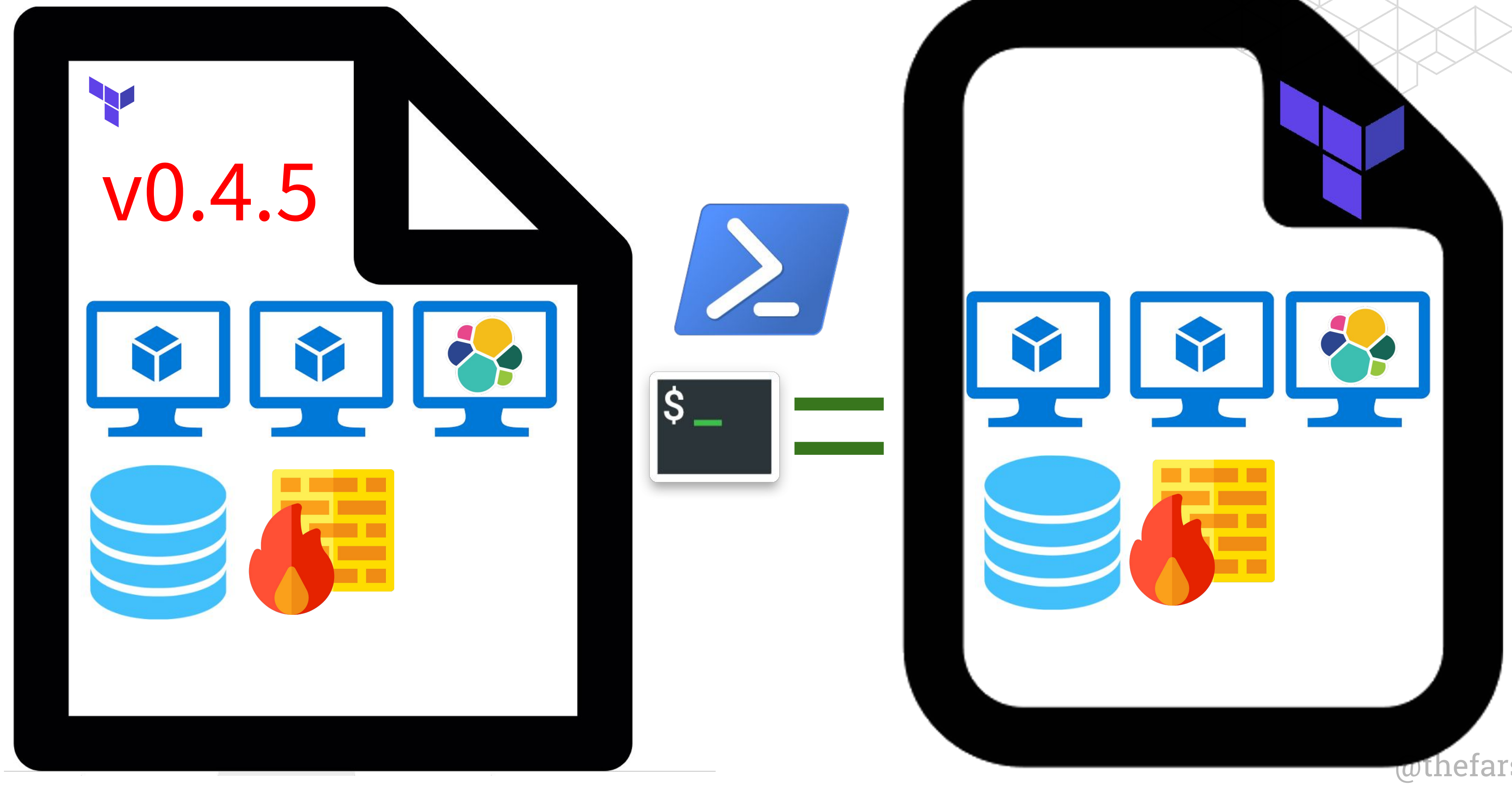


main.tf

```
module "soe" {
  source = "git::ssh://git@enterprise.ds.stackexchange.com/Ent/module.git?ref=v0.4.5"
```

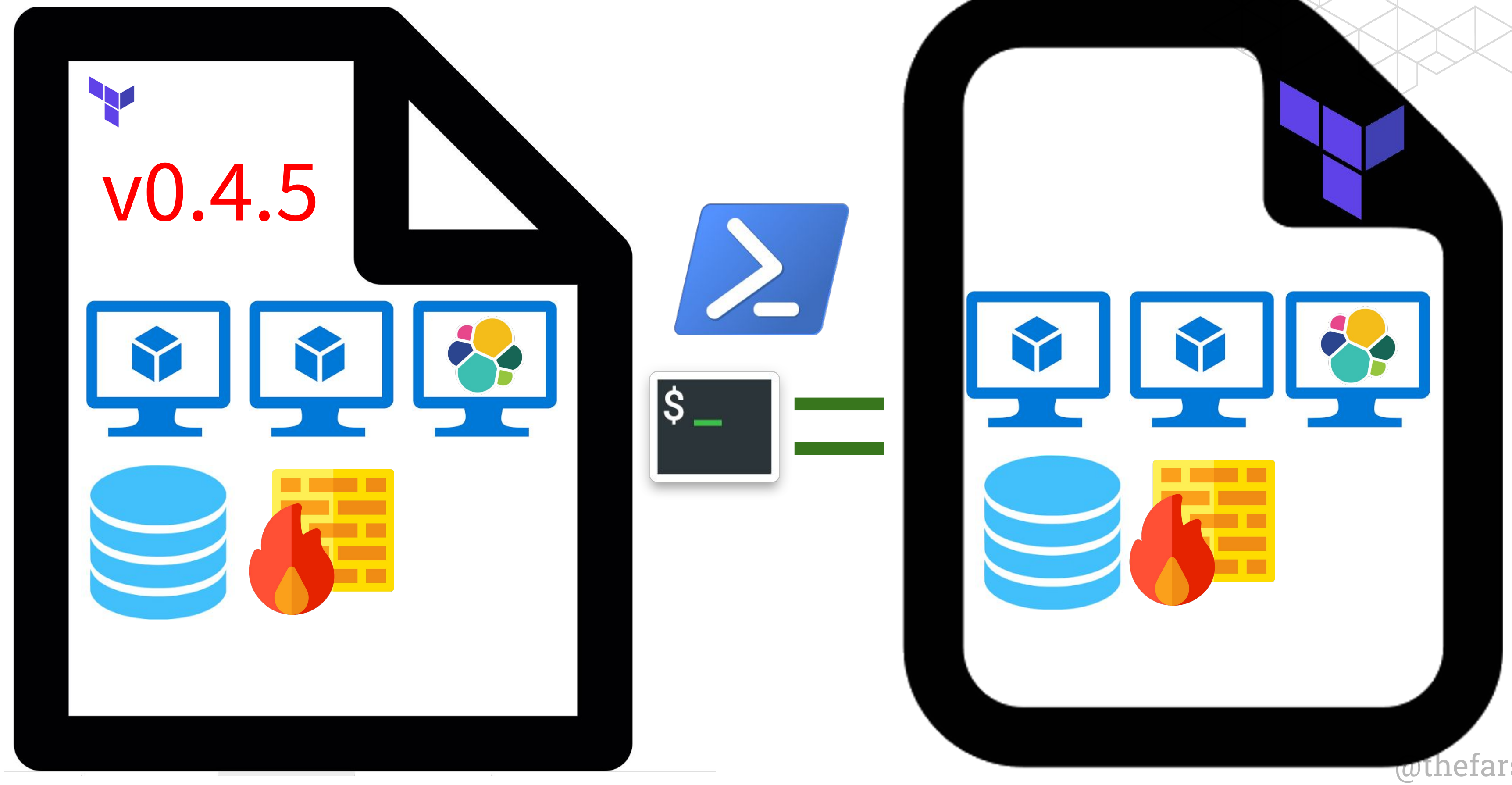
Terraform Module

State File



Terraform Module

State File



Thanks!

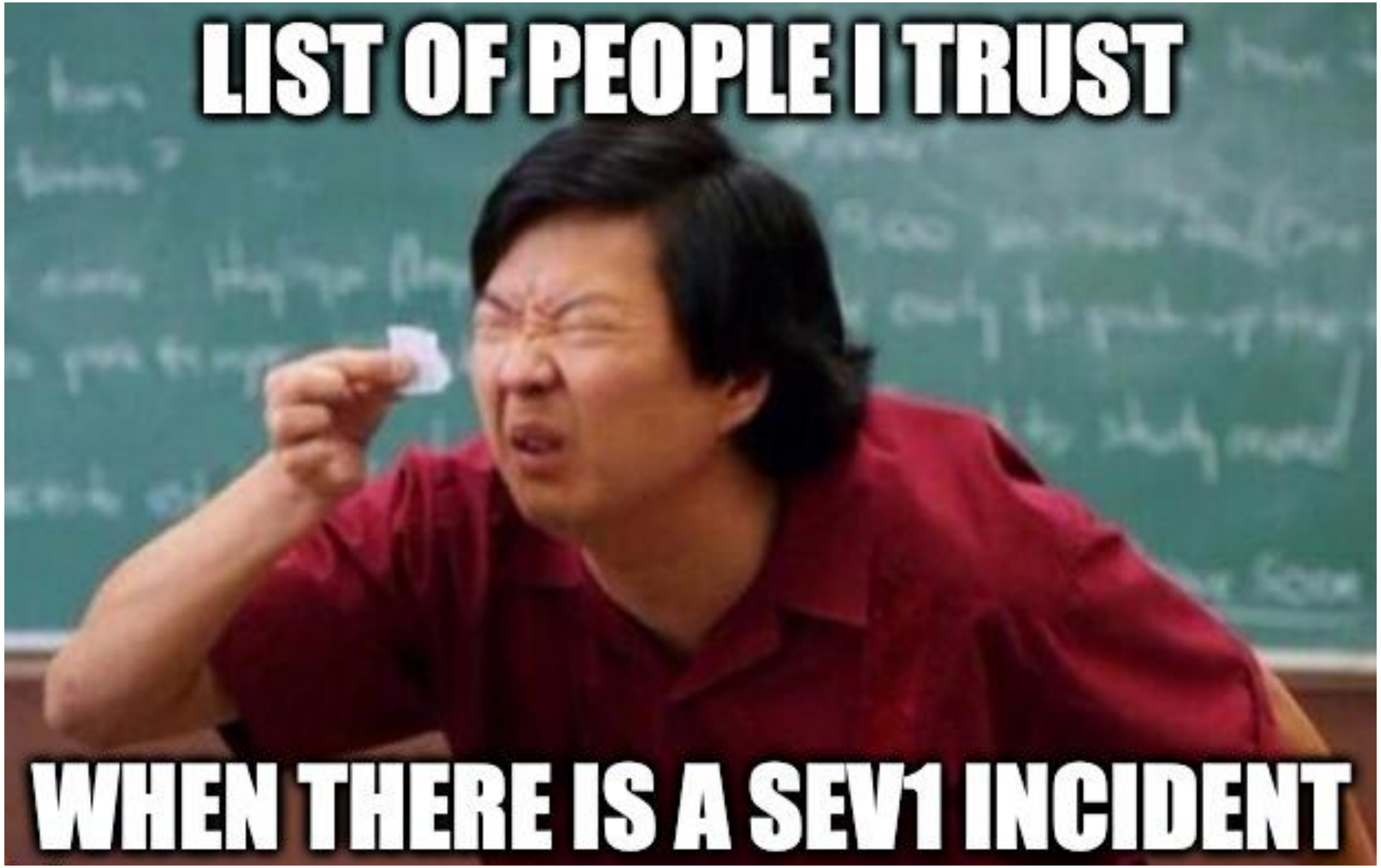
Mark Henderson

@thefarseeker

squadcast

Transparency in Incident Response

How much is too much?


LIST OF PEOPLE I TRUST
WHEN THERE IS A SEV1 INCIDENT

Reliability through obscurity

Reliability through obscurity

Reliability through transparency

squadcast

1 Internal transparency A (Engg team)

squadcast

1 Internal transparency A (Engg team)

2 Internal transparency B (Across entire org)

1. Internal transparency A (Engg team)

2. Internal transparency B (Across entire org)

3. External Transparency A (Customers, Vendors, Stakeholders)

1. Internal transparency A (Engg team)

2. Internal transparency B (Across entire org)

3. External Transparency A (Customers, Vendors, Stakeholders)

4. External Transparency B / Global Transparency (Public)

squadcast

1 Internal transparency A (Engg team)

2 Internal transparency B (Across entire org)

3 External Transparency A (Customers, Vendors, Stakeholders)

4 External Transparency B / Global Transparency (Public)

squadcast

Transparency is central to SRE

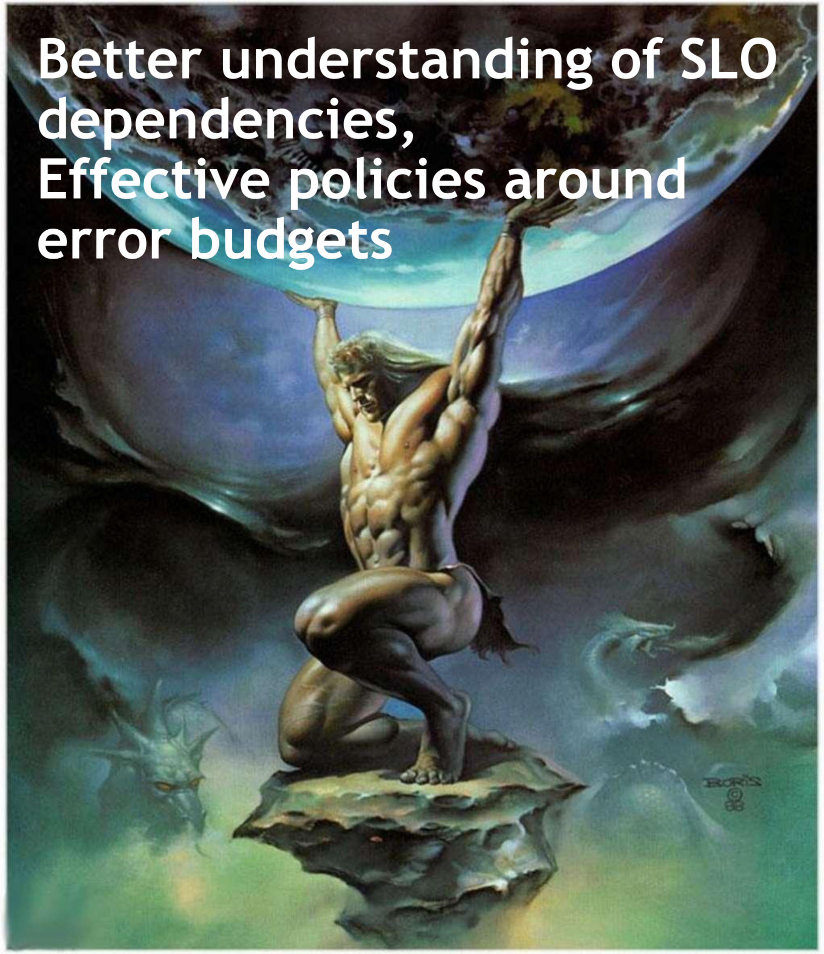
Choosing SLIs, SLOs and improving upon them

squadcast

Transparency is central to SRE

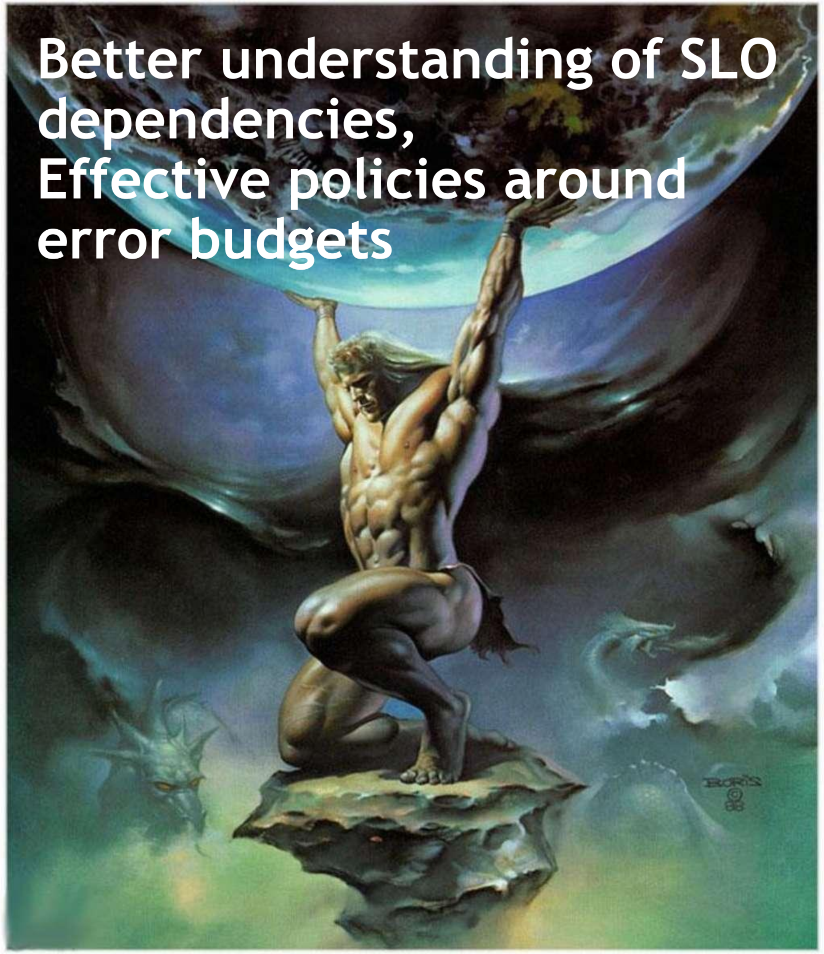
Choosing SLIs, SLOs and improving upon them

Transparency is central to SRE



Better understanding of SLO dependencies,
Effective policies around error budgets

Transparency is central to SRE



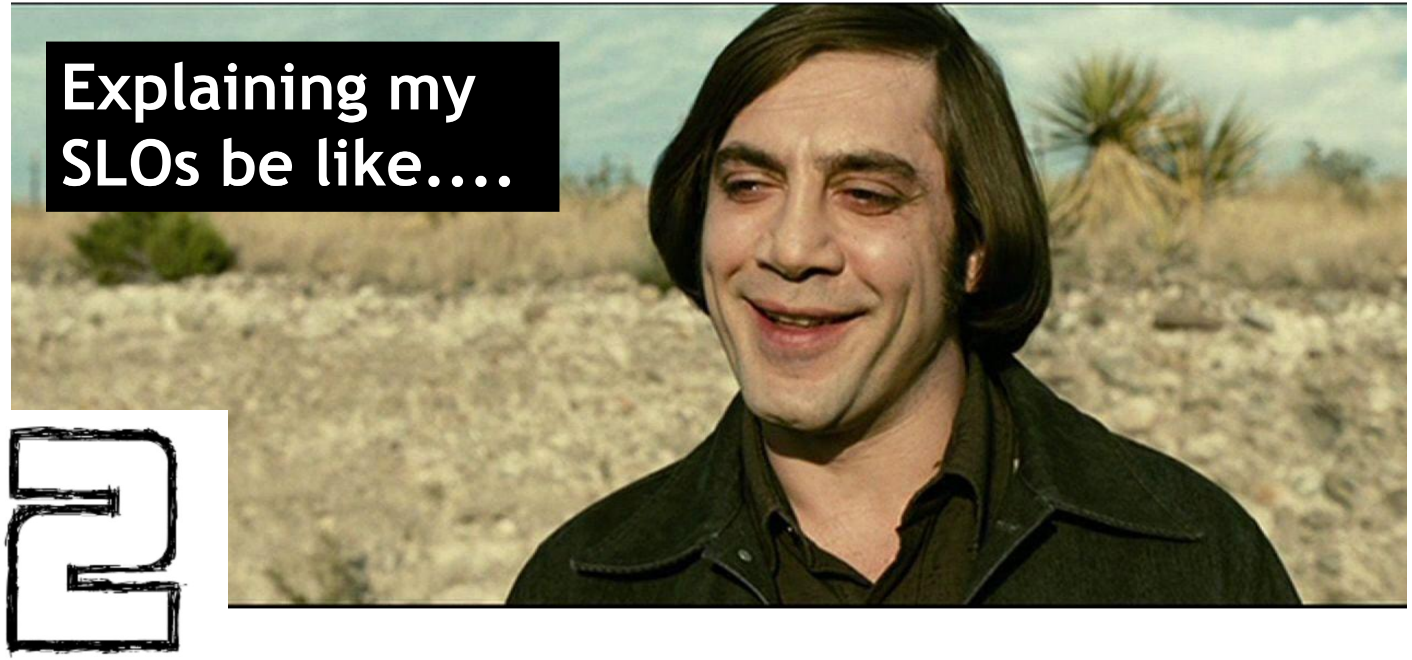
Better understanding of SLO dependencies,
Effective policies around error budgets

Busting Myths

1

squadcast

Busting Myths



squadcast

Busting Myths

squadcast

Benefits of DevOps & SRE Transparency

Simplify Incident Management

Eliminate Toil

Uphold & Track SLOs

Ensure Balanced On-call

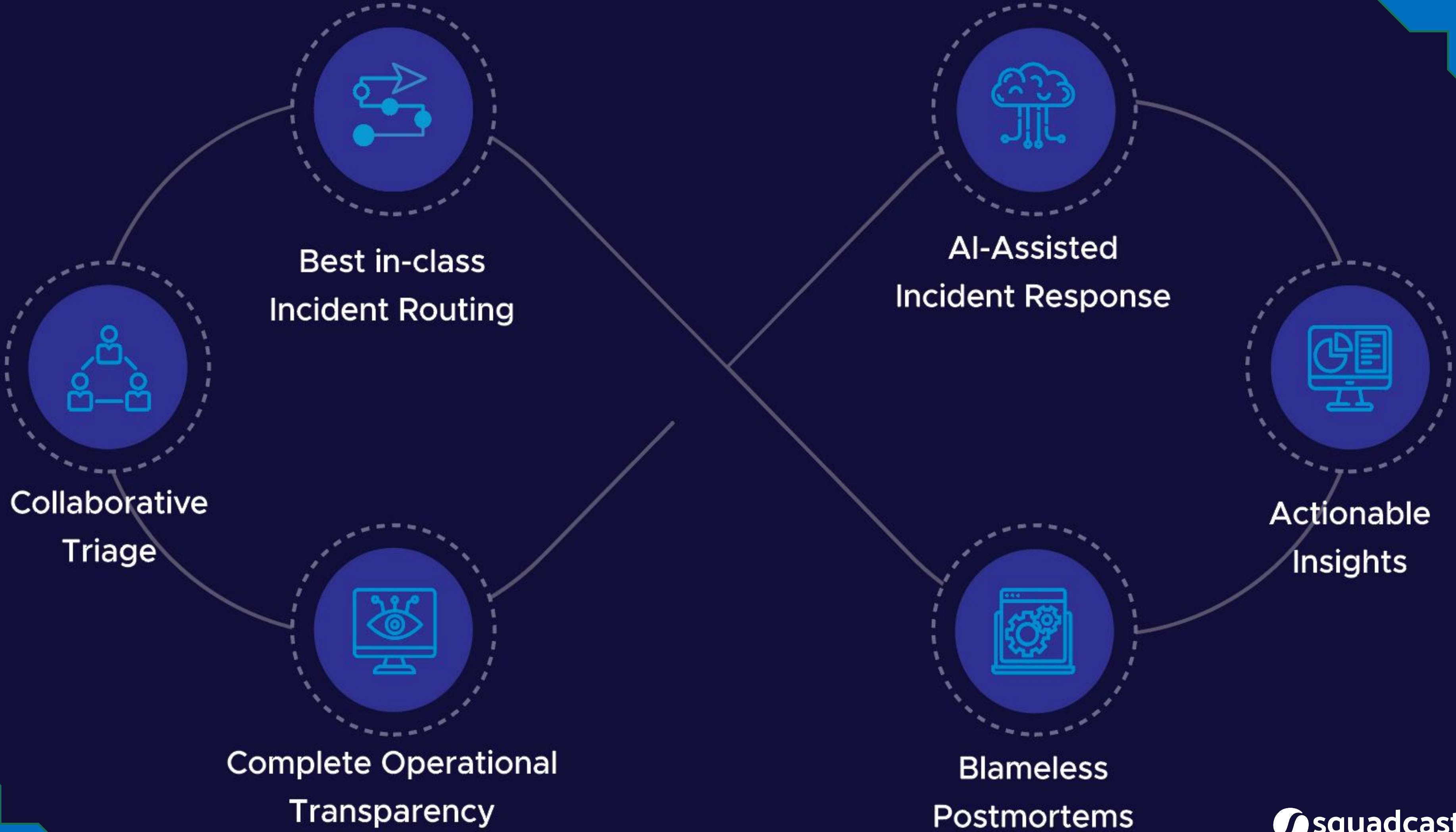
Automate Response

Reduce Alert Fatigue

Increase Operational Transparency

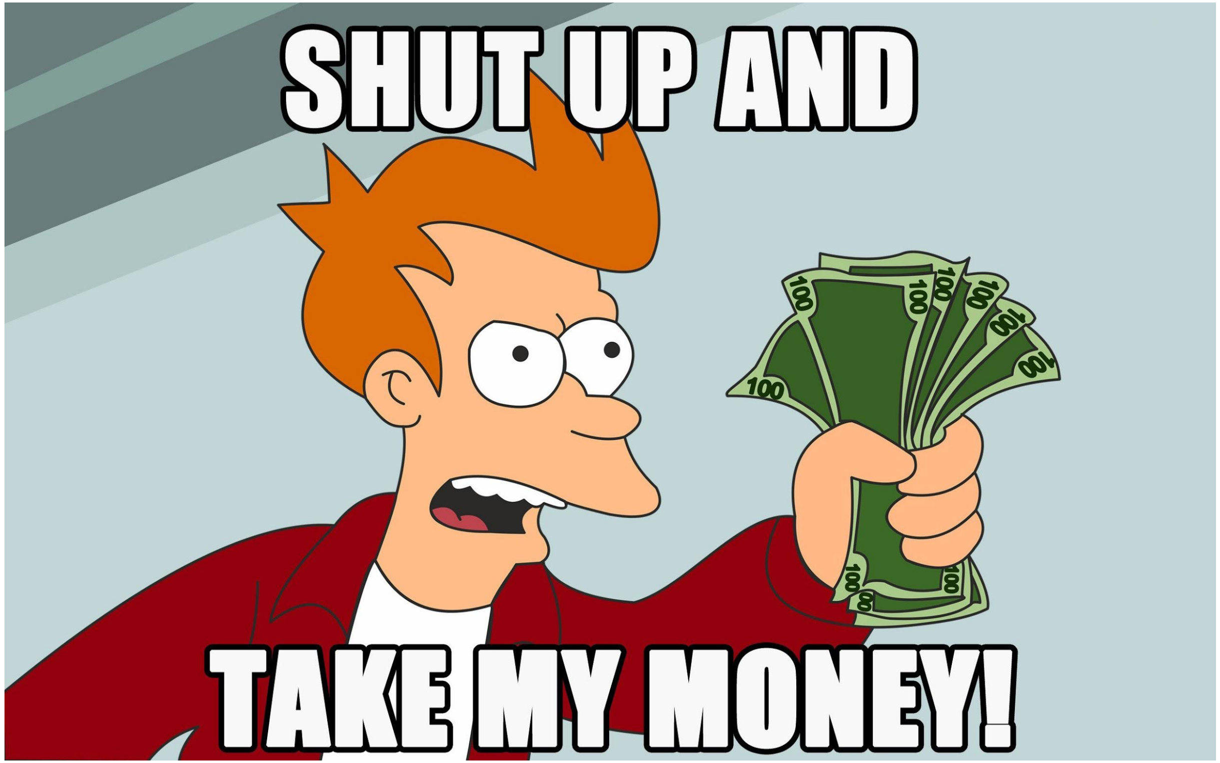
Conduct Blameless Postmortems

squadcast


Best in-class
Incident Routing
AI-Assisted
Incident Response
Collaborative
Triage
Actionable
Insights
Complete Operational
Transparency
Blameless
Postmortems

squadcast

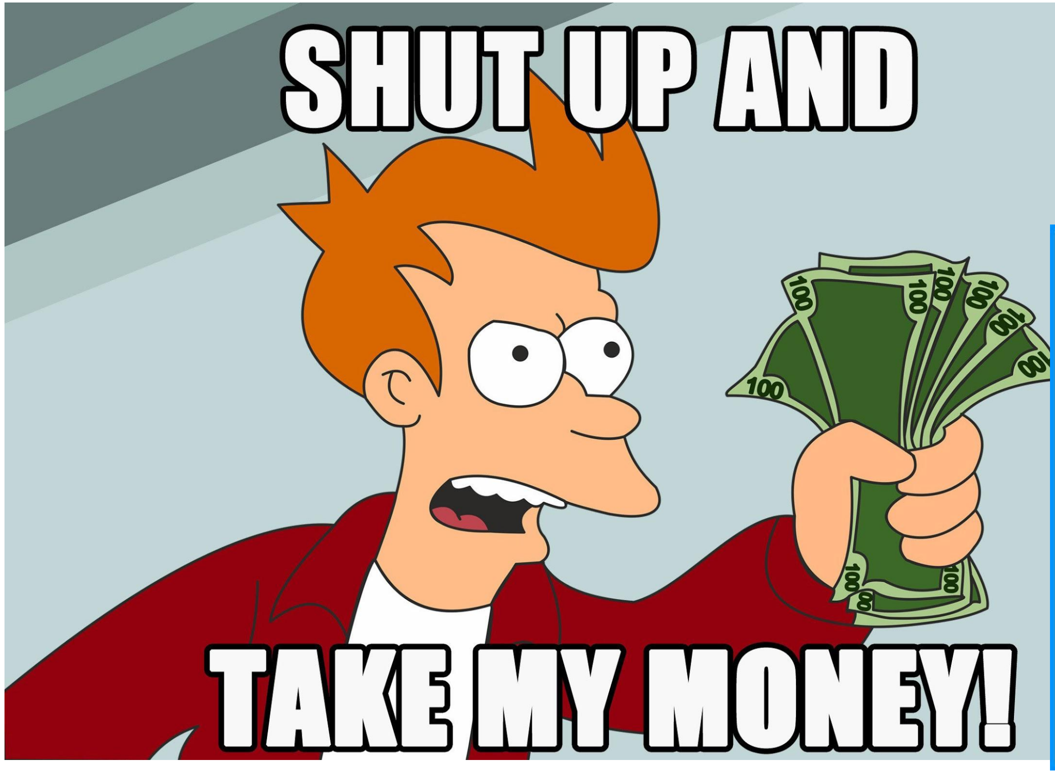
squadcast



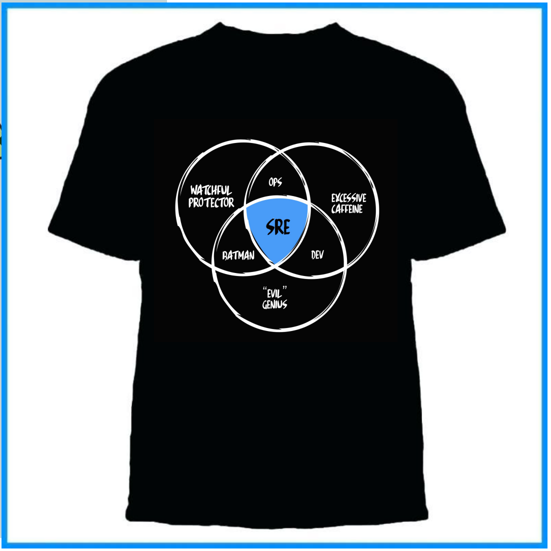
Here's a $500 discount code only for you beautiful people

"SRECONTRANSPARENCY"

squadcast



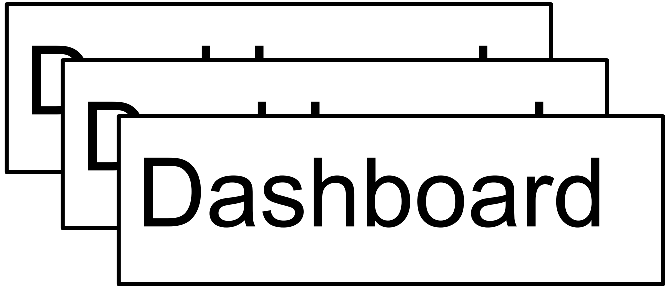
Meet us at Booth #6 for some awesome Swag.



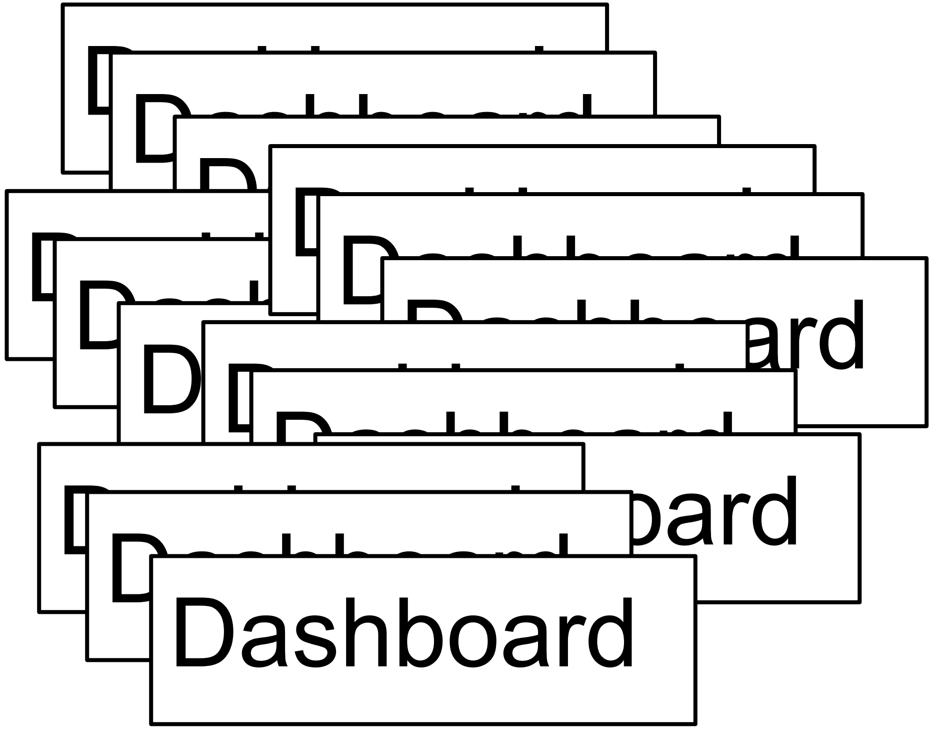
Here's a $500 discount code only for you beautiful people

"SRECONTRANSPARENCY"

“Normal” Company



Google



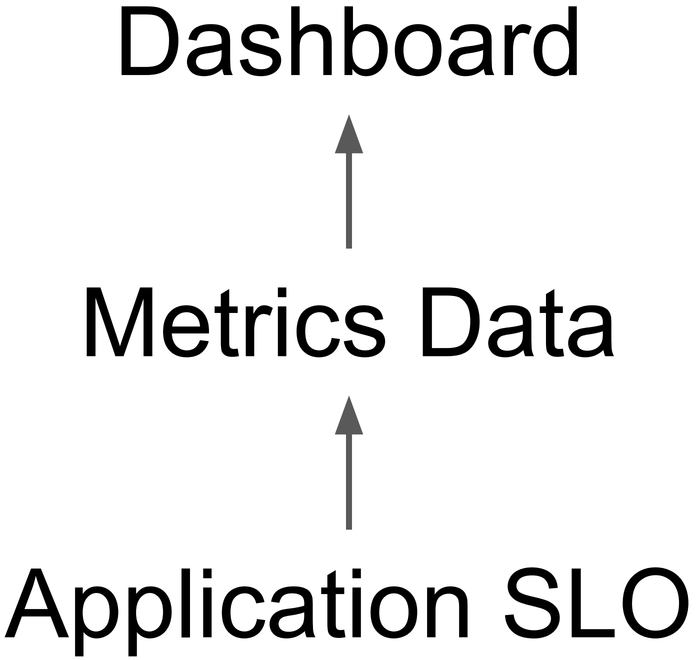
Application SLO

Service Level Objective

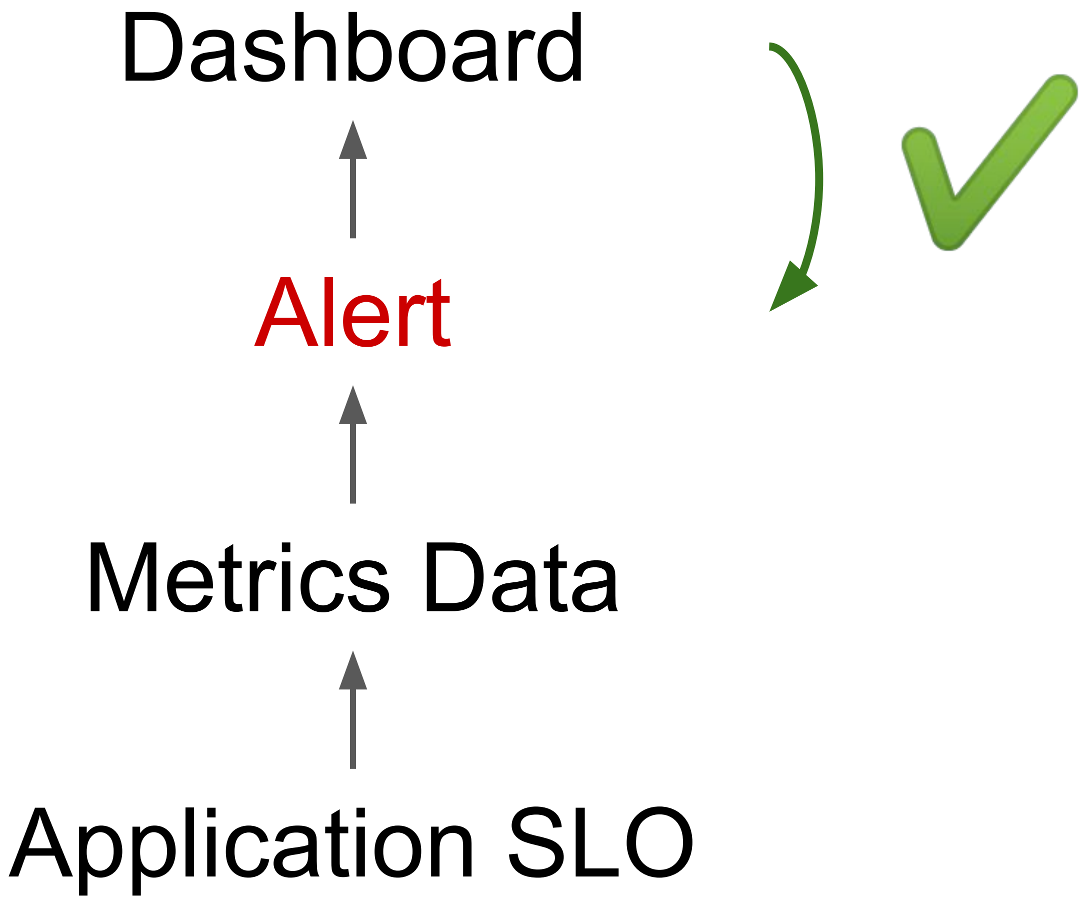
Application SLO

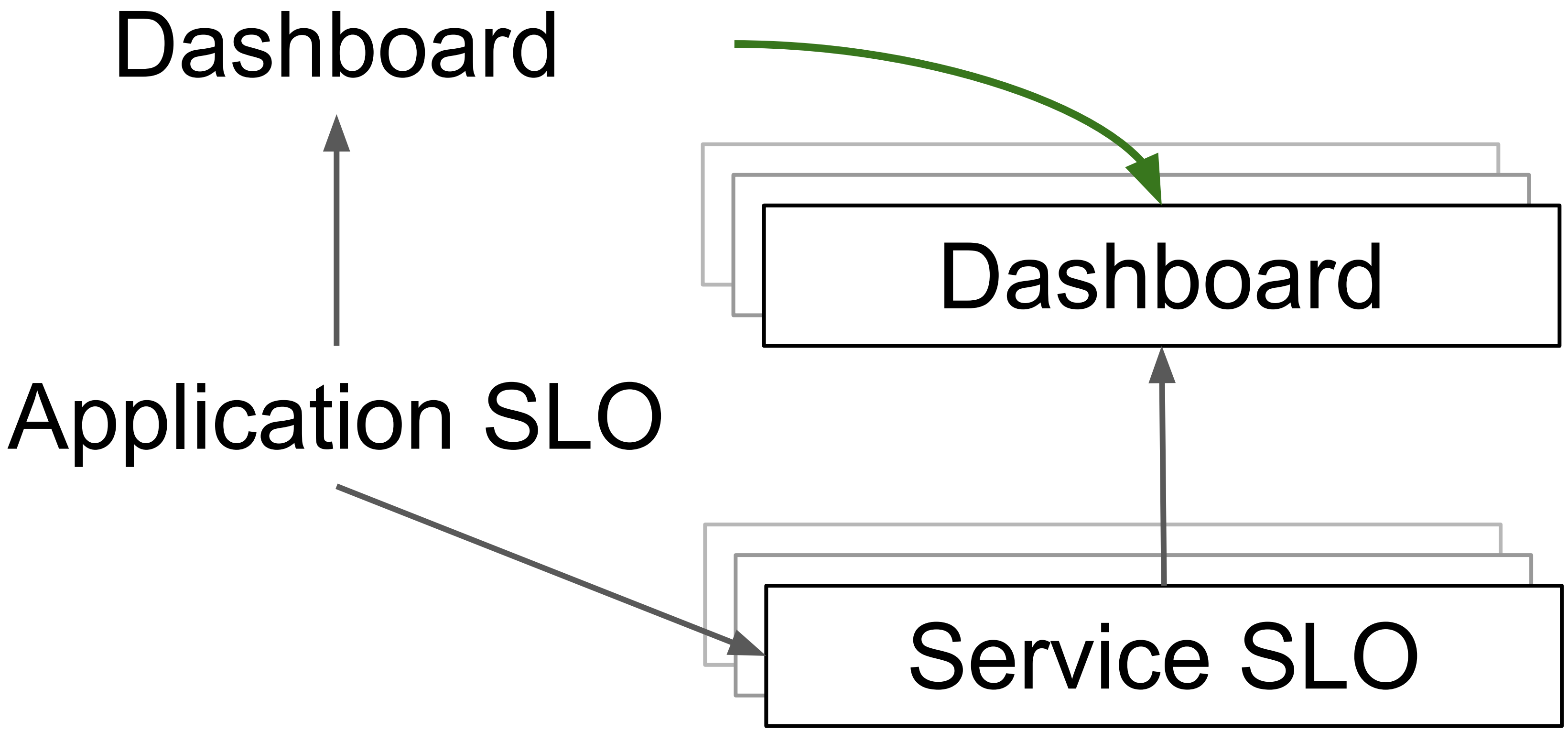
Service Level Objective

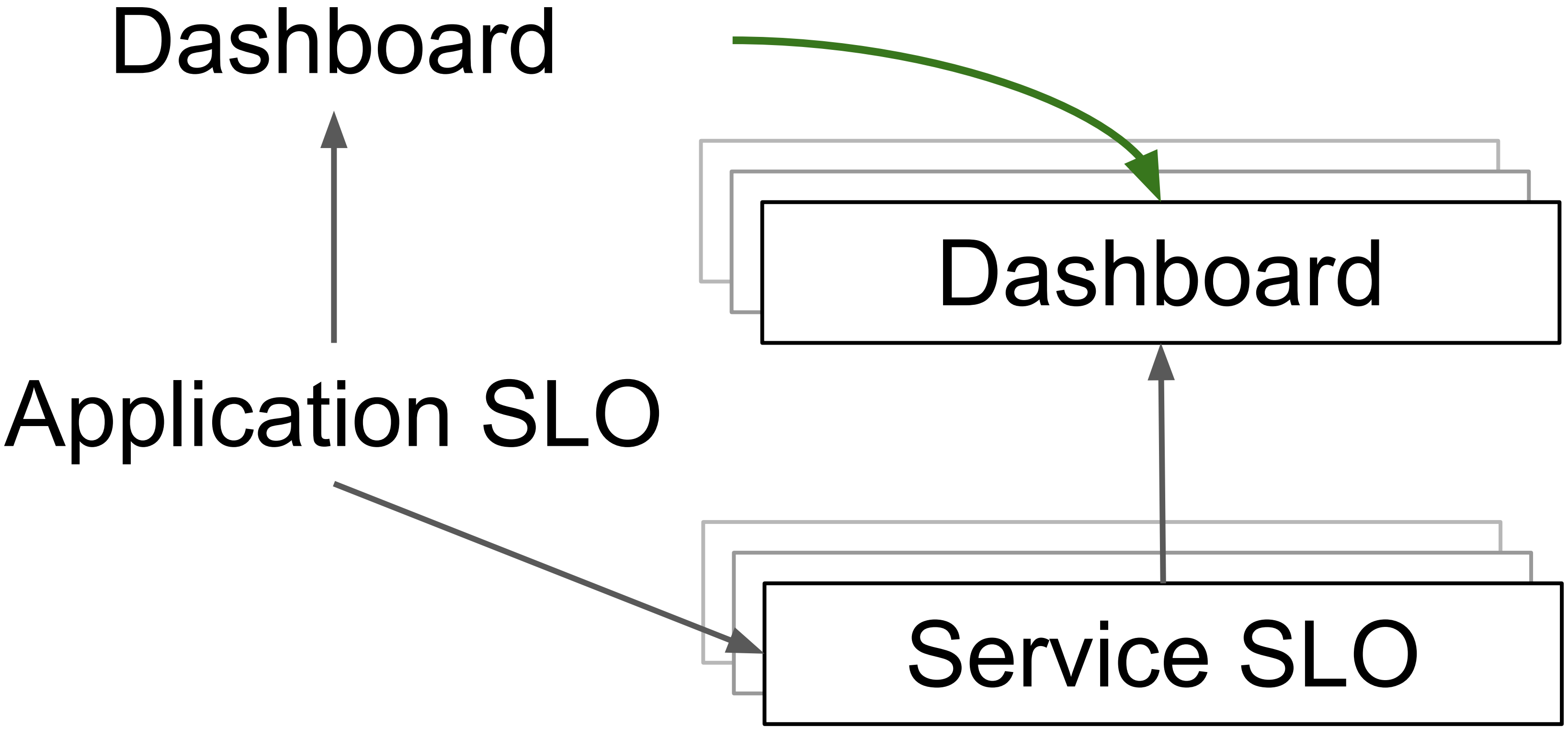
$$\begin{tikzcd} \text{Dashboard} \\ \text{Metrics Data} \arrow[u] \\ \text{Application SLO} \arrow[u] \end{tikzcd}$$


Dashboard
Metrics Data
Application SLO

$$\begin{tikzcd}
\text{Dashboard} \arrow[d, bend left=40, "\checkmark"] \\
\text{Alert} \arrow[u] \\
\text{Metrics Data} \arrow[u] \\
\text{Application SLO} \arrow[u]
\end{tikzcd}$$


Dashboard
Alert
Metrics Data
Application SLO


Dashboard
Application SLO
Dashboard
Service SLO


Dashboard
Application SLO
Dashboard
Service SLO

Standardize

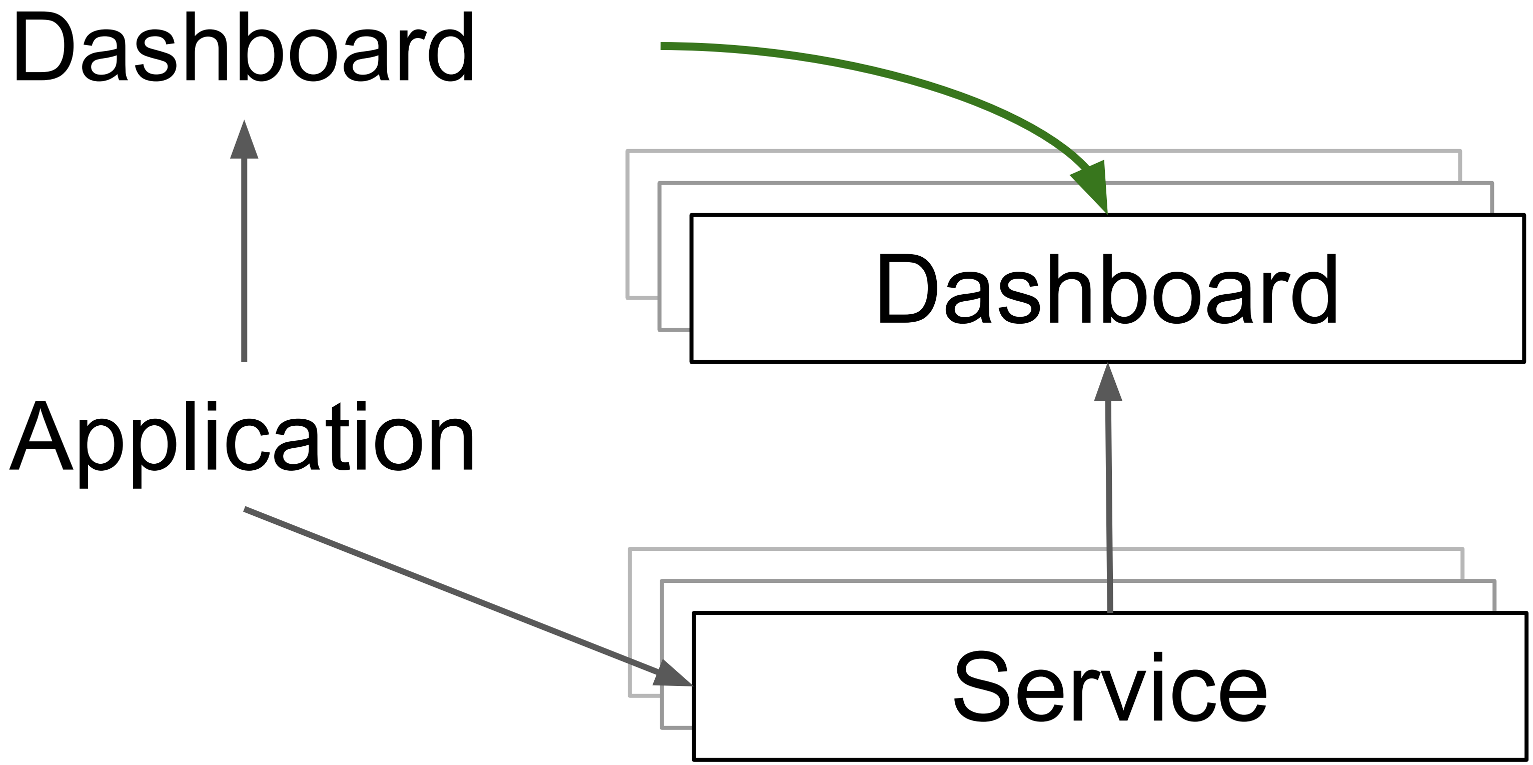
Standardize

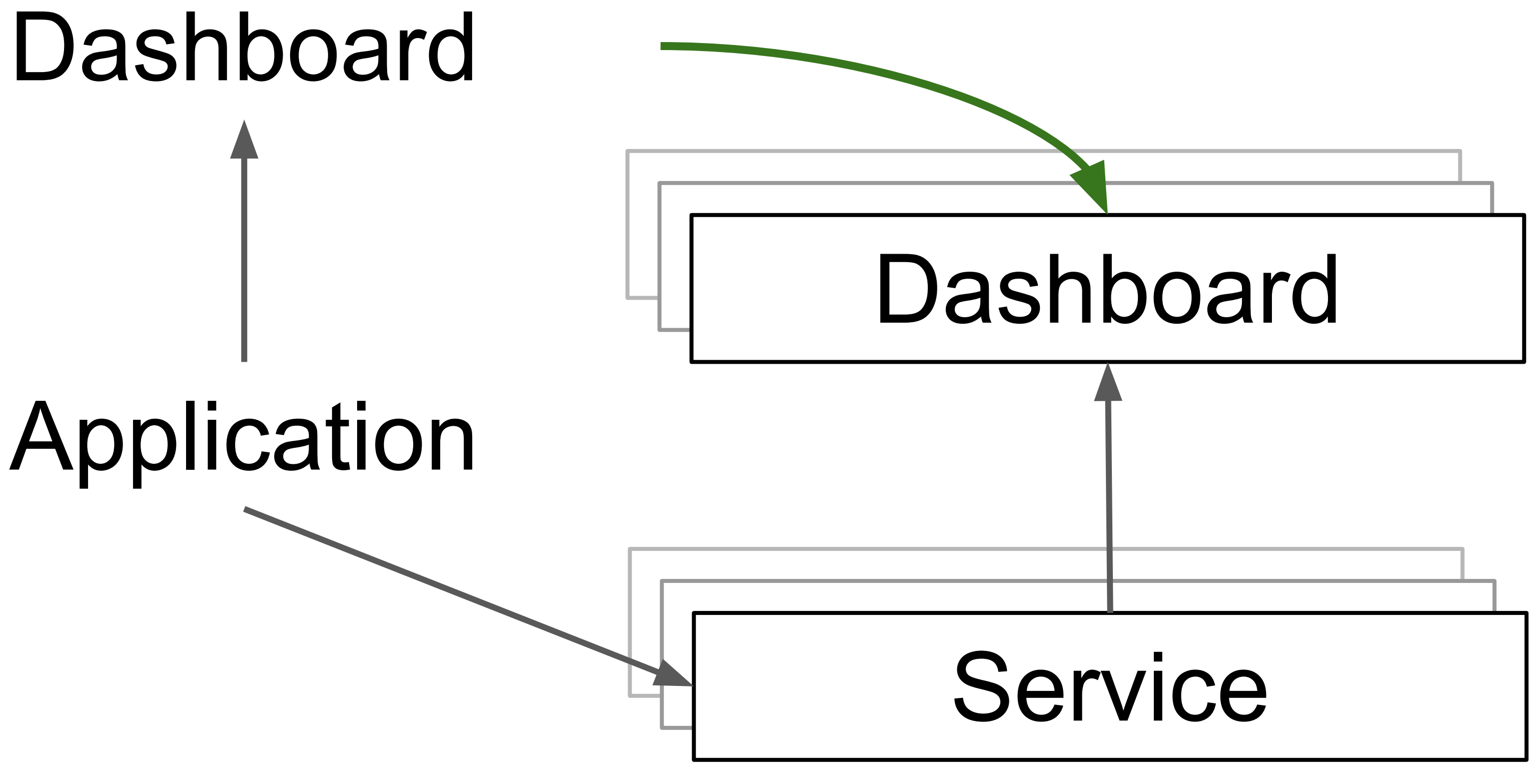
Collect Horizontally

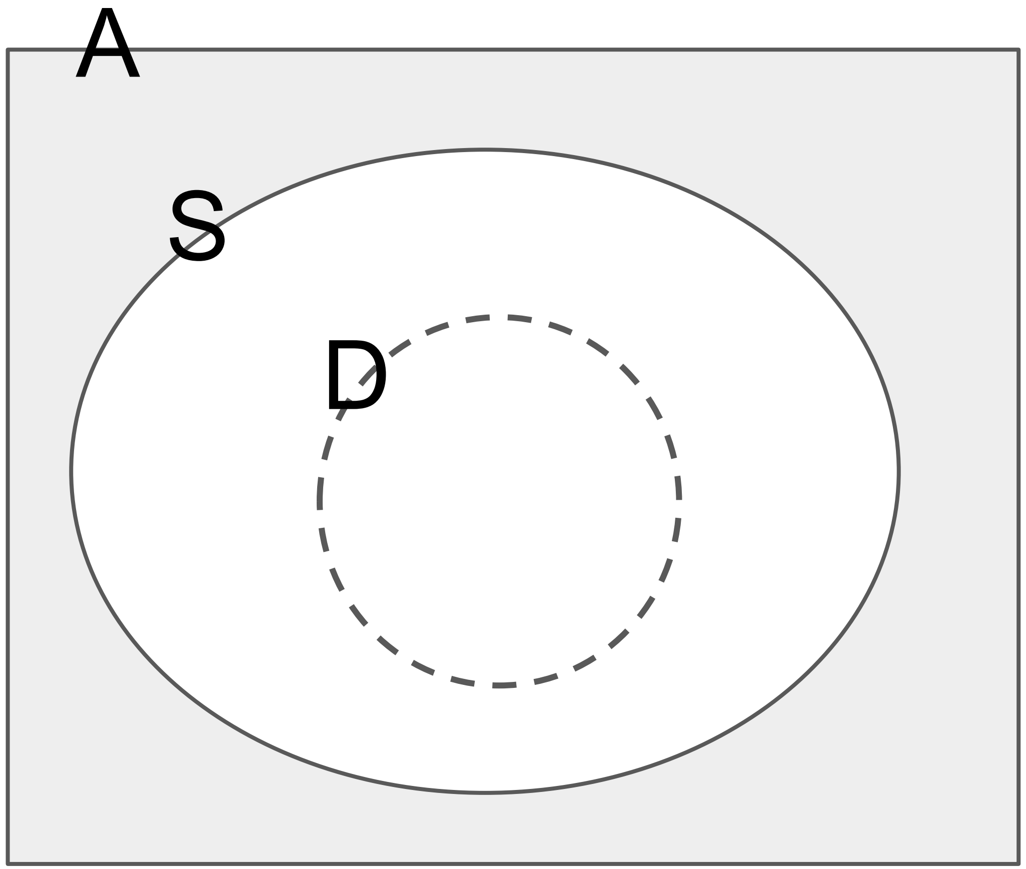
Collect Horizontally

Service Owner
create their
Dashboards

Service Owner create their Dashboards


Dashboard
Application
Dashboard
Service

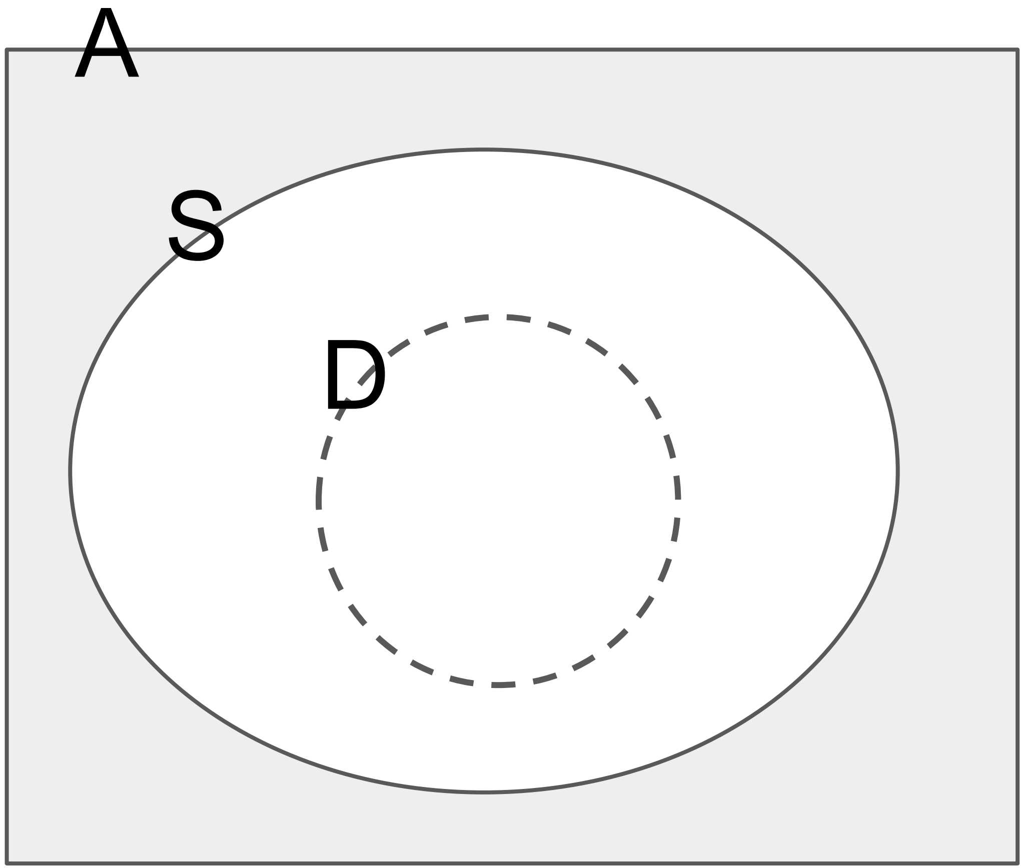

Dashboard
Application
Dashboard
Service



A = All the possible services, jobs and dashboards in your world

S = the Scope of your possible world you’re currently potentially interested in

D = the Drilldown into your dashboards, services and jobs



A = All the possible services, jobs and dashboards in your world

S = the Scope of your possible world you’re currently potentially interested in

D = the Drilldown into your dashboards, services and jobs

Thank You

@bobak

Preventative Paradigm
What SREs can learn from Moms

Rayappa Mayakunthala
@mrayappa

About Me

Director of Software Engineering

Hyderabad, India

All opinions expressed are solely my own and do not express the views or opinions of Salesforce.

Pictures are not mine and the respective owners may have copyrights.

شكراً جزيلاً

thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
dziękuję
bedankt
спасибо
рахмат
Баярлалаа
mersi
kia ora
barka
welalin
tack
spas
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
хвала
asante
manana
obrigada
tenki
murakoze
chokrane
kiitos
dankie
nanni
nandri
dhanyavad
hvala
mauruuru
köszönöm
gràcie
bayarlalaa
enkosi
aciu
dankon
akun
djiere dieuf
tau
mochchakkeram
дякую
mamnun
chnorakaloutioun
gratias ago
gràcies
sulpáy
go raibh maith agat
sobodi
dėkuji
sagolun
sukriya
kop khun krap
taiku
arigatô
takk
dakujem
trugarez
mèsi
didi madloba
ありがとう
najis tuke
tanemirt
grazie
shukriya
mercé
мерси
kam sah hamnida
rahmat
terima kasih
rahmet
diolch
dhanyavadagalu
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ

About Topic

How Moms nurture happy kids the Preventative way has some important lessons for the SREs.

We can build, scale and run healthy distributed systems by leveraging the same principles.

Preventative by design - Nurturing Happy Kids

Reliability

- Effective Monitoring
- Being on-call
- Tracking outages

Reliability

- Effective Monitoring
- Being on-call
- Tracking outages

Reliability

- Effective Monitoring
- Being on-call
- Tracking outages

Security

- Embrace Risk
- Address cascading failures
- integrity

Security

- Embrace Risk
- Address cascading failures
- integrity

Security

- Embrace Risk
- Address cascading failures
- integrity

Scalability

- Pipelines
- Blameless RCA
- Config Management

Scalability

- Pipelines
- Blameless RCA
- Config Management

Scalability

- Pipelines
- Blameless RCA
- Config Management

Performance

- SLO's
- Manage Load
- Eliminate Toil
- Learn/Teach/Mentor

Performance

- SLO's
- Manage Load
- Eliminate Toil
- Learn/Teach/Mentor

Performance

- SLO's
- Manage Load
- Eliminate Toil
- Learn/Teach/Mentor

Preventative by design - Nurturing Happy Kids

Preventative by design - Nurturing Happy Kids

SRE needs more women engineers

Our Practices of Delegating Ownership in Microservices World

Daisuke Fujita (@dtan4)
Mercari, Inc.
Microservices Platform Team

June 14, 2019
SREcon19 Asia/Pacific Lightning Talk

mercari

delegating ownership?

At Mercari Microservices

mercari

Monolith (-2018)

Good at first, but not scalable

- 🔺 Complexity
- 🔻 Velocity

Need to change the system & organization, before growing up more

Monolith (-2018)

Good at first, but not scalable

- 🔺 Complexity
- 🔻 Velocity

Need to change the system & organization, before growing up more

mercari

Microservices at Mercari

Scale the organization & Maximize output

👉 small / autonomous / cross-functional teams

👉 strong ownership by the teams

Microservices at Mercari

Scale the organization & Maximize output

👉 small / autonomous / cross-functional teams

👉 strong ownership by the teams

Ownership by Microservice Team

Service teams
write code, deploy it, run it
by themselves

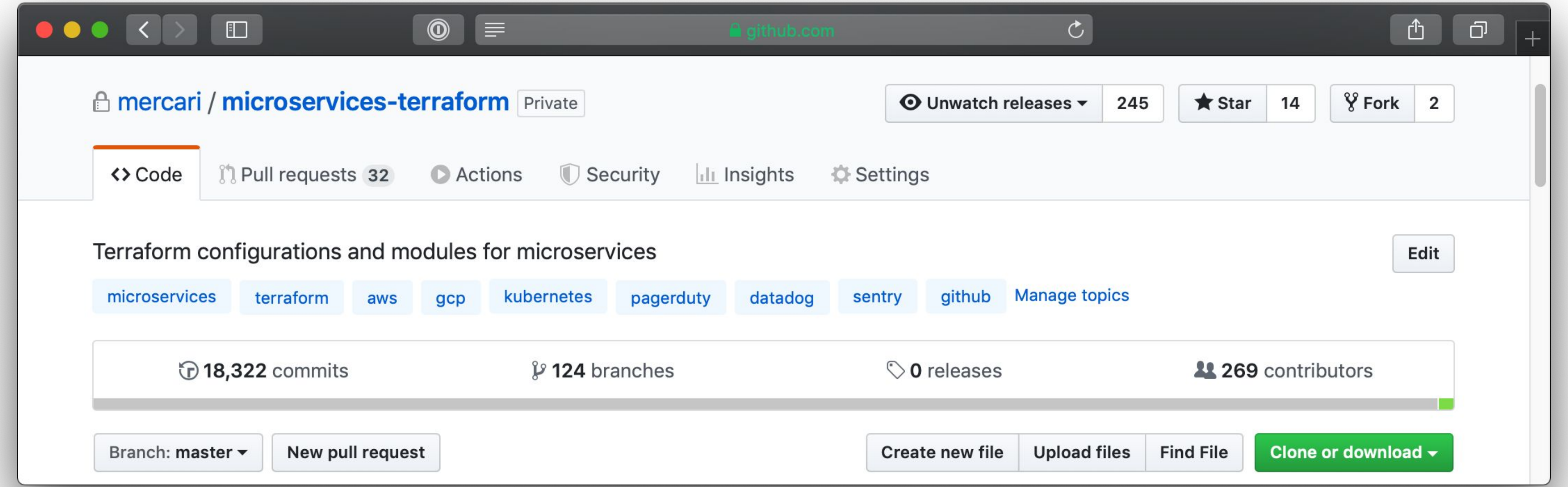
Ownership by Microservice Team

**Service teams
write code, deploy it, run it
by themselves**

…how? 🤔

Q: How to prepare microservice's infrastructure?

Monorepo for Terraform Configurations



github.com

mercari / microservices-terraform Private

Unwatch releases 245 | Star 14 | Fork 2

Code | Pull requests 32 | Actions | Security | Insights | Settings

Terraform configurations and modules for microservices

Edit

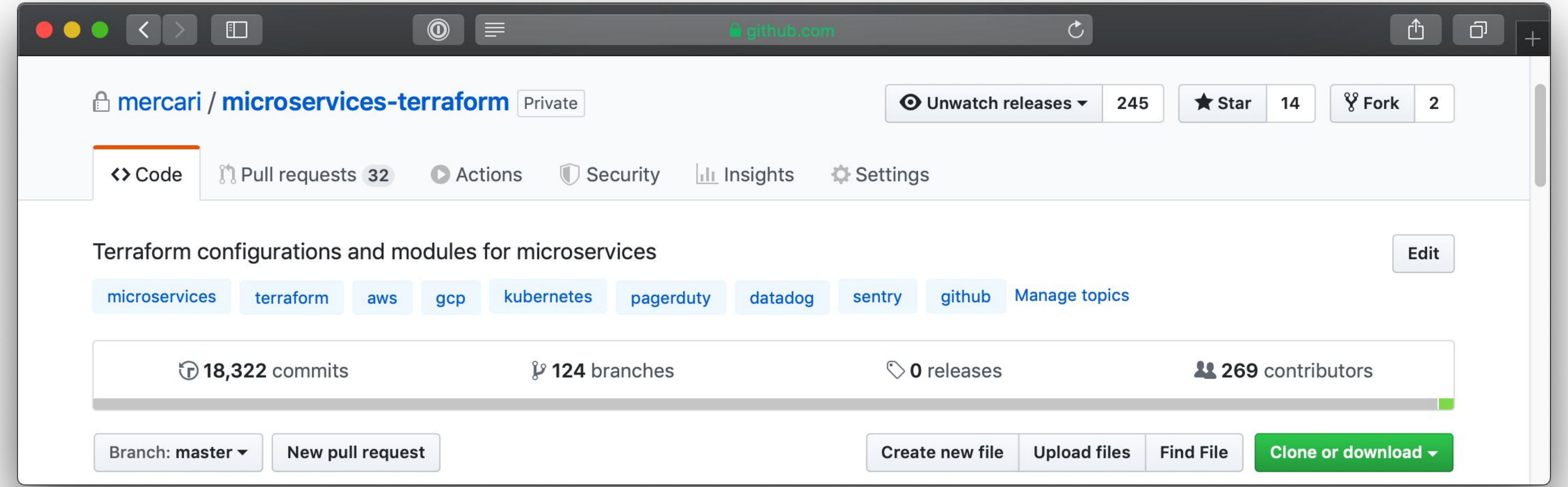
microservices terraform aws gcp kubernetes pagerduty datadog sentry github Manage topics

18,322 commits | 124 branches | 0 releases | 269 contributors

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

mercari

Monorepo for Terraform Configurations


github.com

mercari / microservices-terraform Private

Unwatch releases 245 | Star 14 | Fork 2

Code | Pull requests 32 | Actions | Security | Insights | Settings

Terraform configurations and modules for microservices

Edit

microservices terraform aws gcp kubernetes pagerduty datadog sentry github Manage topics

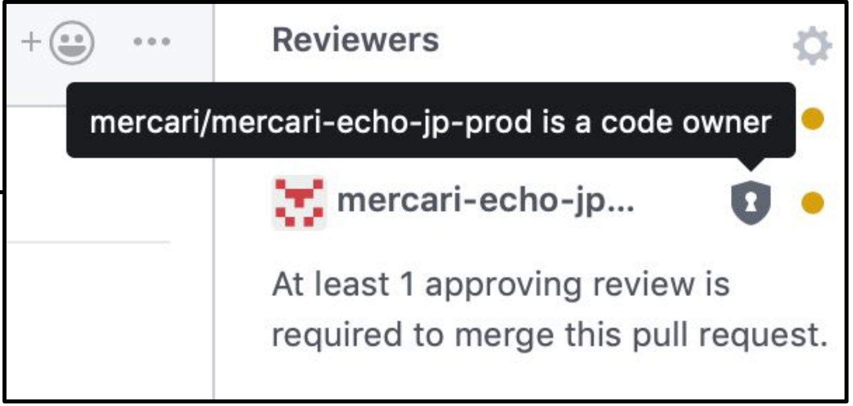
18,322 commits | 124 branches | 0 releases | 269 contributors

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

mercari

Monorepo for Terraform Configurations

```
terraform/microservices
|-- mercari-echo-jp
|    |-- development
|    `-- production
|         |-- backend.tf
|         |-- google_bigquery_dataset.tf
|         |-- google_spanner_database.tf
|         |-- module_microservice_starter_kit.tf
|         |-- providers.tf
|         `-- variables.tf
`-- mercari-listing-jp
     |-- development
     `-- production
```


Reviewers

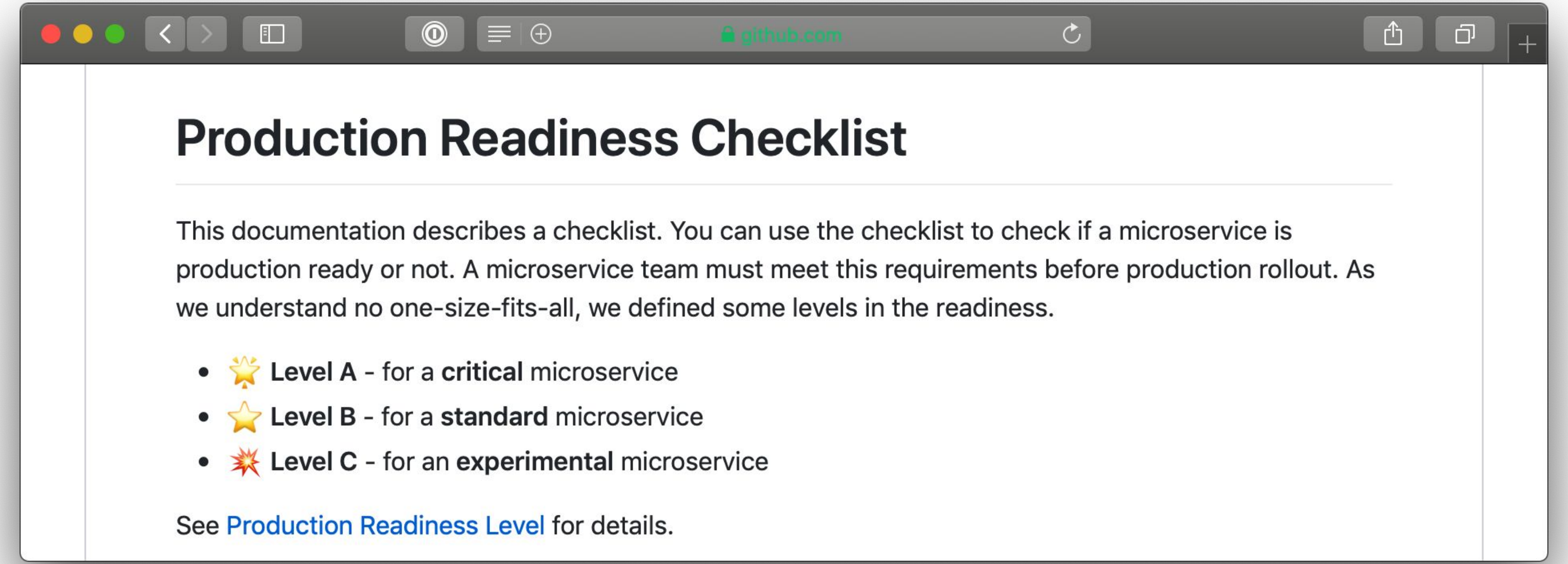
mercari/mercari-echo-jp-prod is a code owner

mercari-echo-jp...

At least 1 approving review is required to merge this pull request.

Q: What are the microservices requirements in production?

Production Readiness Checklist



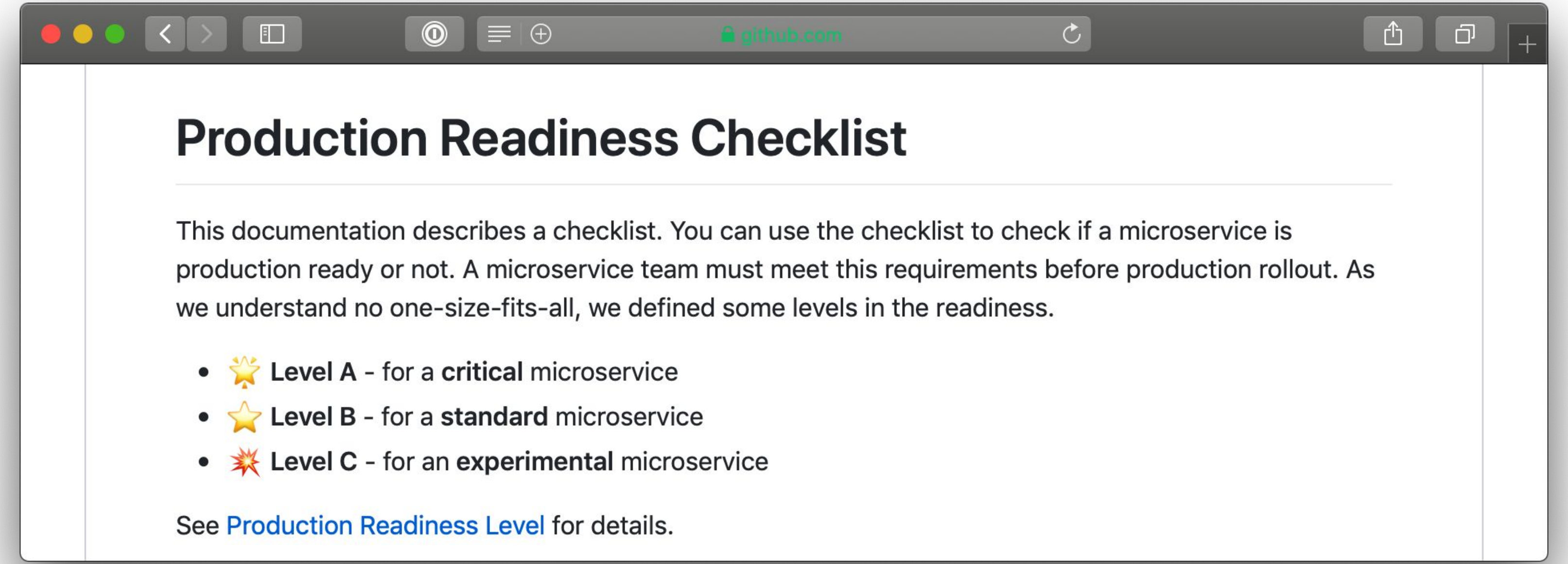
Production Readiness Checklist

This documentation describes a checklist. You can use the checklist to check if a microservice is production ready or not. A microservice team must meet this requirements before production rollout. As we understand no one-size-fits-all, we defined some levels in the readiness.

- 🌟 **Level A** - for a **critical** microservice
- ⭐ **Level B** - for a **standard** microservice
- 💥 **Level C** - for an **experimental** microservice

See Production Readiness Level for details.

Production Readiness Checklist



Production Readiness Checklist

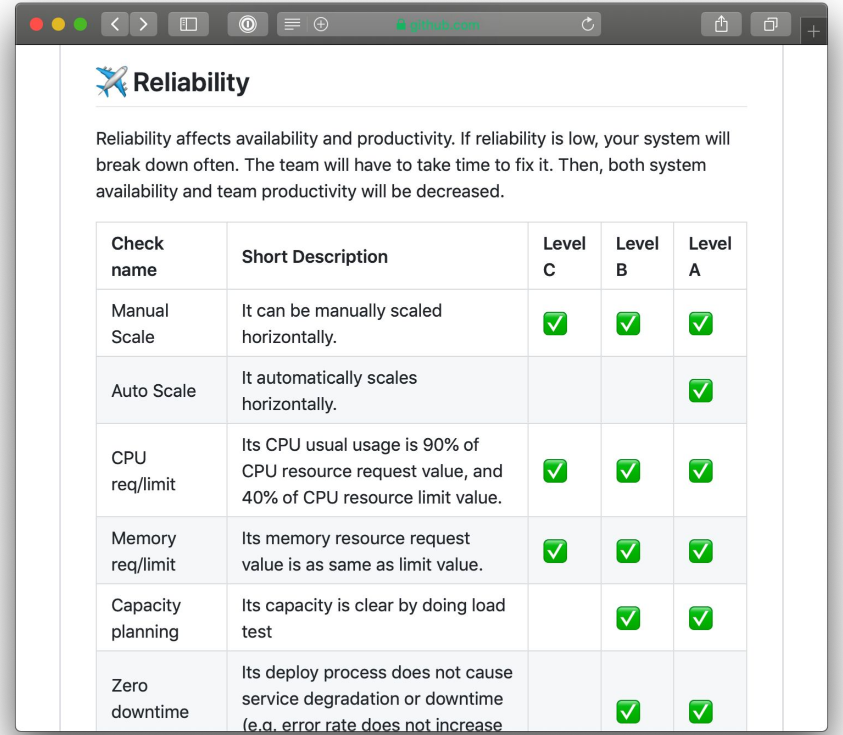
This documentation describes a checklist. You can use the checklist to check if a microservice is production ready or not. A microservice team must meet this requirements before production rollout. As we understand no one-size-fits-all, we defined some levels in the readiness.

- 🌟 **Level A** - for a **critical** microservice
- ⭐ **Level B** - for a **standard** microservice
- 💥 **Level C** - for an **experimental** microservice

See Production Readiness Level for details.

Production Readiness Checklist

- Maintainability
- Durability
- Observability
- Reliability
- Security
- Accessibility
- Sustainability
- Data Storage



✈️ Reliability

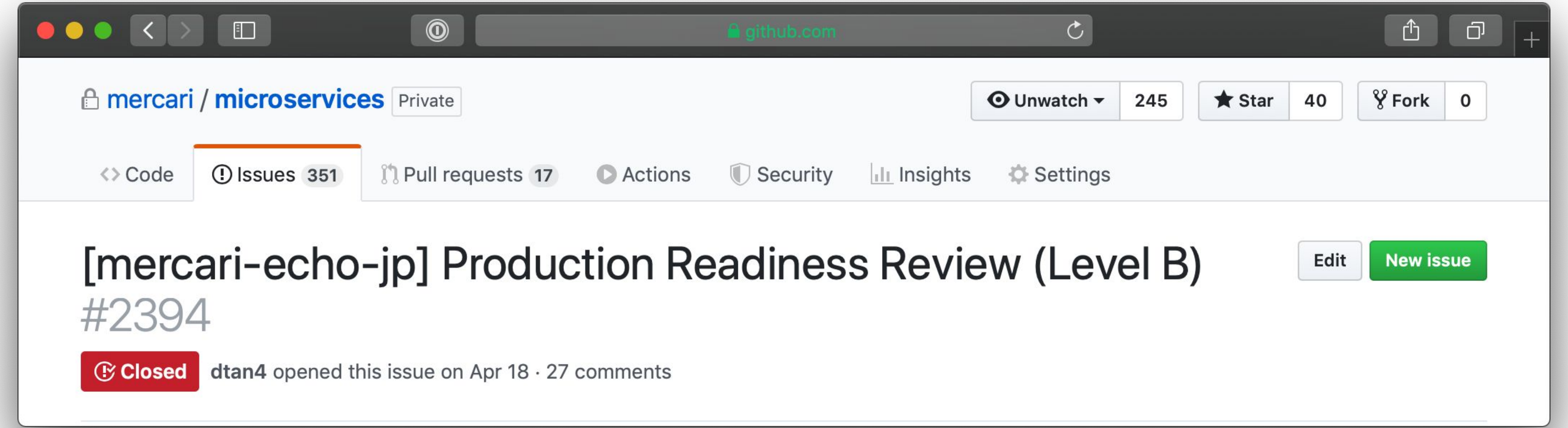
Reliability affects availability and productivity. If reliability is low, your system will break down often. The team will have to take time to fix it. Then, both system availability and team productivity will be decreased.

Check name	Short Description	Level C	Level B	Level A
Manual Scale	It can be manually scaled horizontally.	✅	✅	✅
Auto Scale	It automatically scales horizontally.			✅
CPU req/limit	Its CPU usual usage is 90% of CPU resource request value, and 40% of CPU resource limit value.	✅	✅	✅
Memory req/limit	Its memory resource request value is as same as limit value.	✅	✅	✅
Capacity planning	Its capacity is clear by doing load test		✅	✅
Zero downtime	Its deploy process does not cause service degradation or downtime (e.g. error rate does not increase		✅	✅

mercari

Production Readiness Checklist

GitHub issue-based check & review process


github.com

mercari / microservices Private

Unwatch 245 | Star 40 | Fork 0

Code | Issues 351 | Pull requests 17 | Actions | Security | Insights | Settings

[mercari-echo-jp] Production Readiness Review (Level B) #2394

Edit | New issue

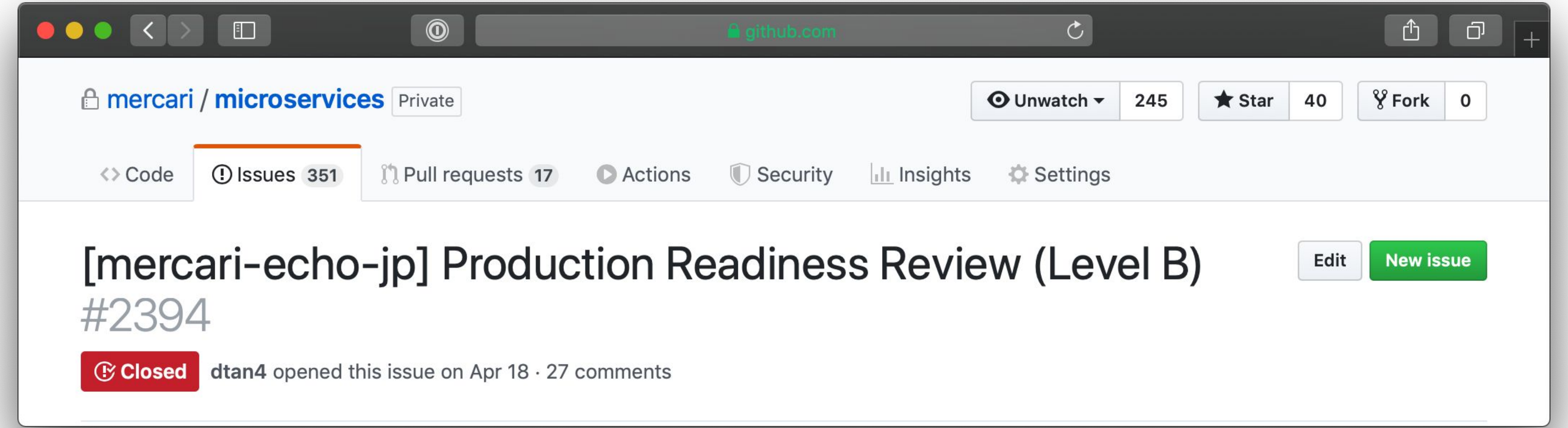
Closed dtan4 opened this issue on Apr 18 · 27 comments

TODO: automated regular check

mercari

Production Readiness Checklist

GitHub issue-based check & review process



github.com

mercari / microservices Private

[mercari-echo-jp] Production Readiness Review (Level B) #2394

Closed dtan4 opened this issue on Apr 18 · 27 comments

TODO: automated regular check

mercari

Conclusion

Ownership by microservice teams

- Monorepo for provisioning microservice infrastructure
- Production Readiness Checklist

It contributes to organizational expansion with microservices

https://careers.mercari.com

Conclusion

Ownership by microservice teams

- Monorepo for provisioning microservice infrastructure
- Production Readiness Checklist

It contributes to organizational expansion with microservices

https://careers.mercari.com