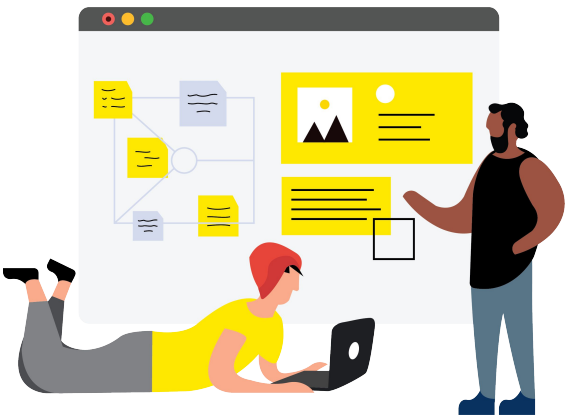
eficode



Practical Implementation of **PoLP** with **RBAC** and **Threat Modelling** in **K8s**

.. and win for "longest title" with most acronyms.

Nicolaj Græsholt

@figaw

eficode

A long time ago...

eficode

about:Nicolaj Græsholt

Consultant ~5y

Speaker

Trainer

@figaw

MSc. Comp.Sci. from Aarhus University in Cryptography

eficode

Agenda

eficode

about:talk

DON'T PANIC

(Large, friendly letters)

Everything You See Is What You Get

https://github.com/figaw/polp-rbac-k8s

eficode

Kubernetes (K8s)

Lights, Camera, Abstraction!

Kubernetes in 30 seconds

open source

Google

CLOUD NATIVE COMPUTING FOUNDATION

Industry Standard

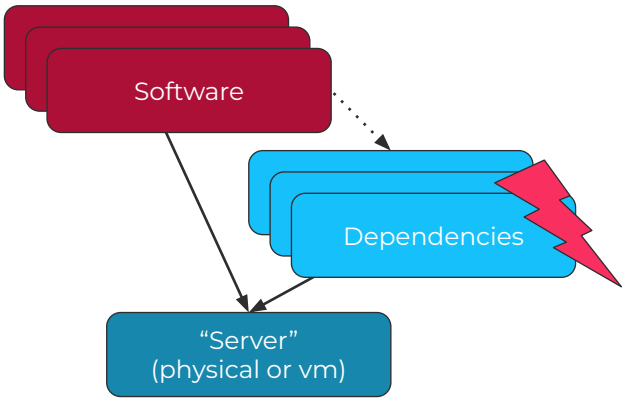
Highly Customizable

kubernetes

”Planet Scale … platform for automating deployment, scaling, and operations of application containers across clusters of hosts.” - kubernetes.io

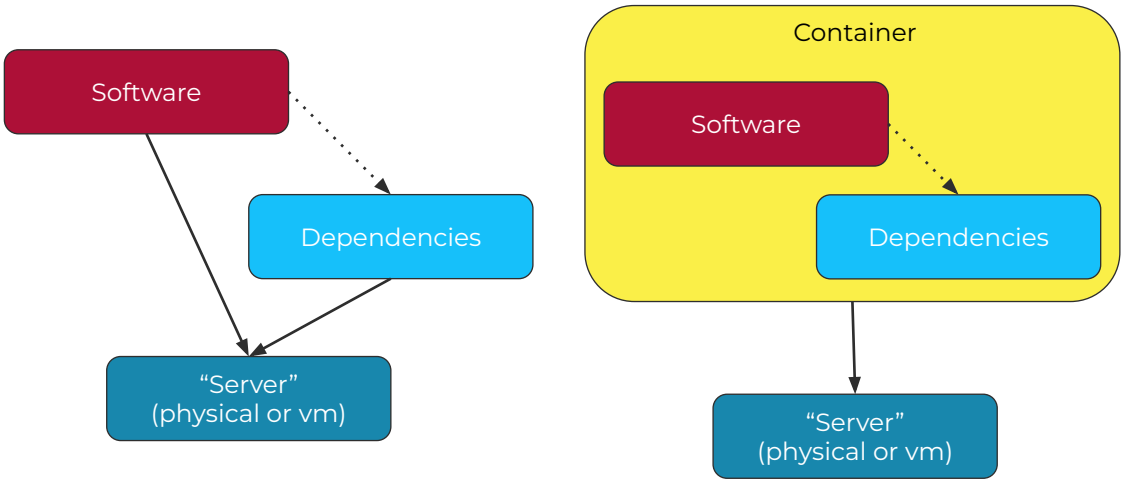
eficode

Kubernetes



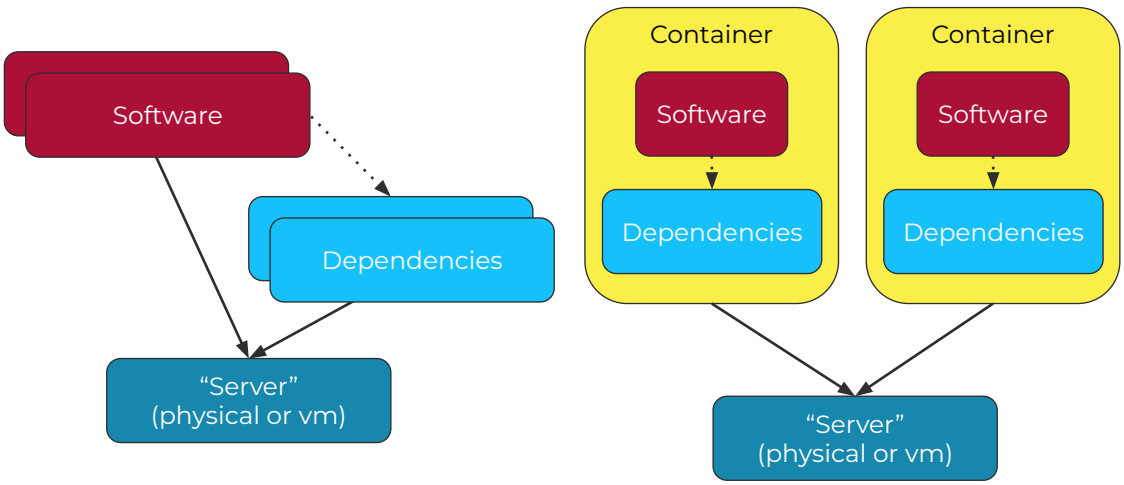
eficode

Kubernetes



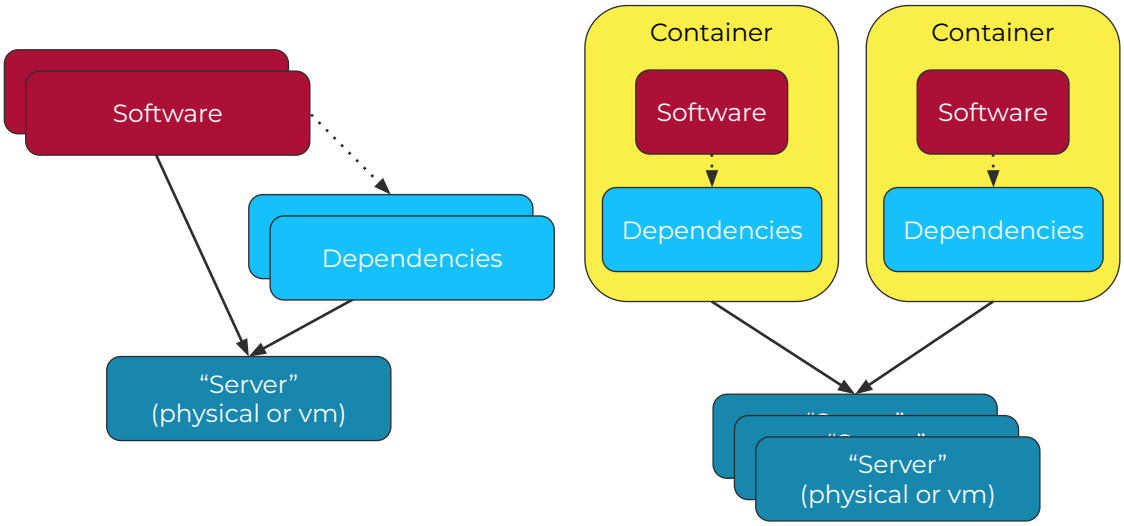
eficode

Kubernetes



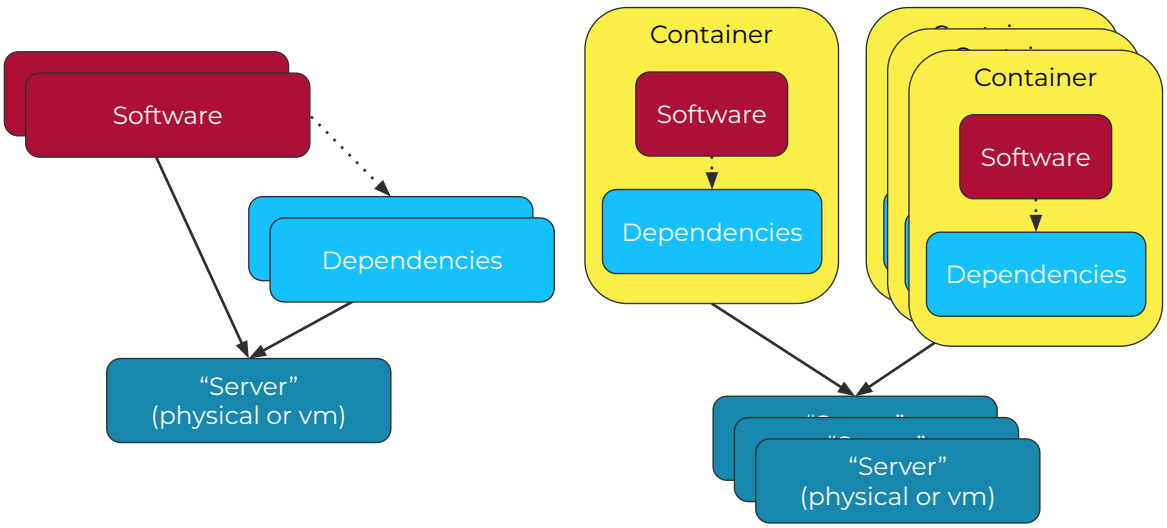
eficode

Kubernetes



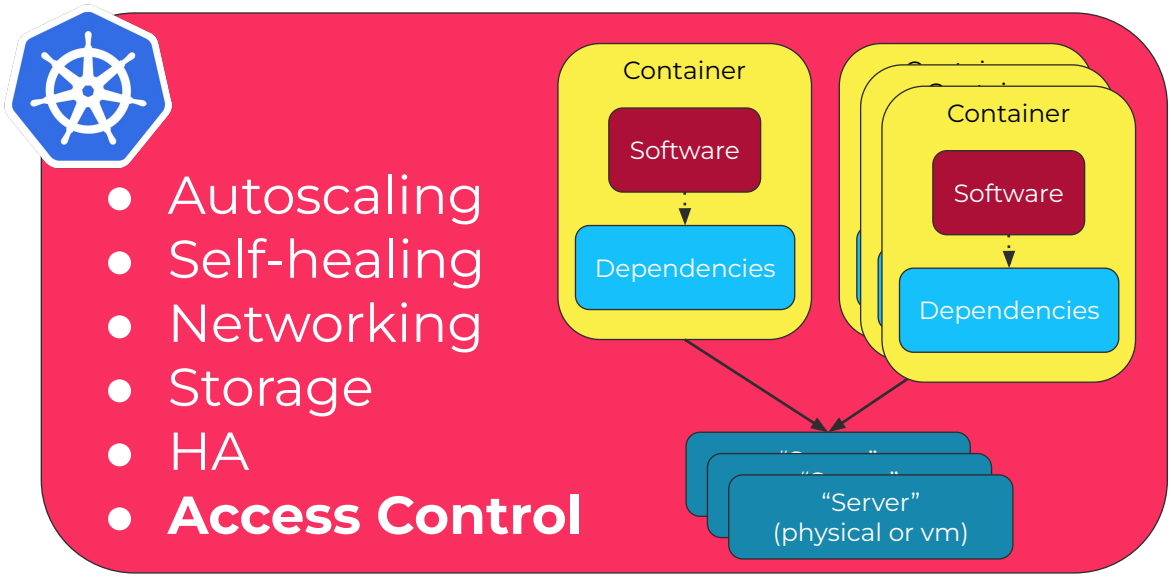
eficode

Kubernetes



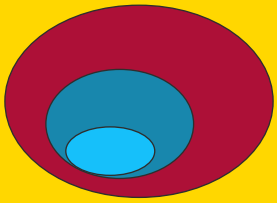
eficode

Kubernetes



- Autoscaling
- Self-healing
- Networking
- Storage
- HA
- **Access Control**

eficode



The Principle of Least Privilege (PoLP)

Yay, vocabulary!

eficode

The Principle of Least Privilege states that ***a subject should be given only those privileges needed for it to complete its task***.

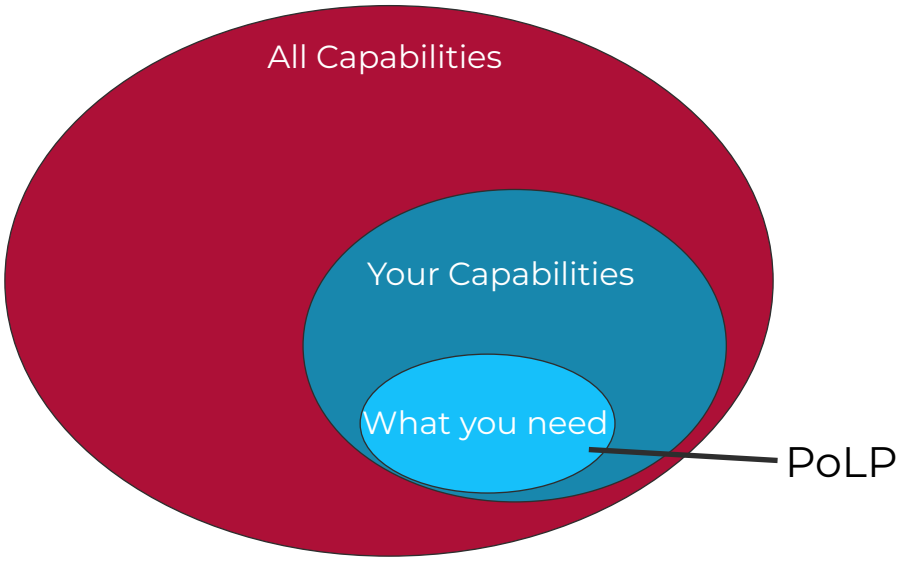
- *Bishop [Bishop 03] in Chapter 13, "Design Principles," Section 13.2.1, "Principle of Least Privilege," pages 343-344,*
  https://www.cisa.gov/uscert/bsi/articles/knowledge/principles/least-privilege

eficode

Venn Diagrams

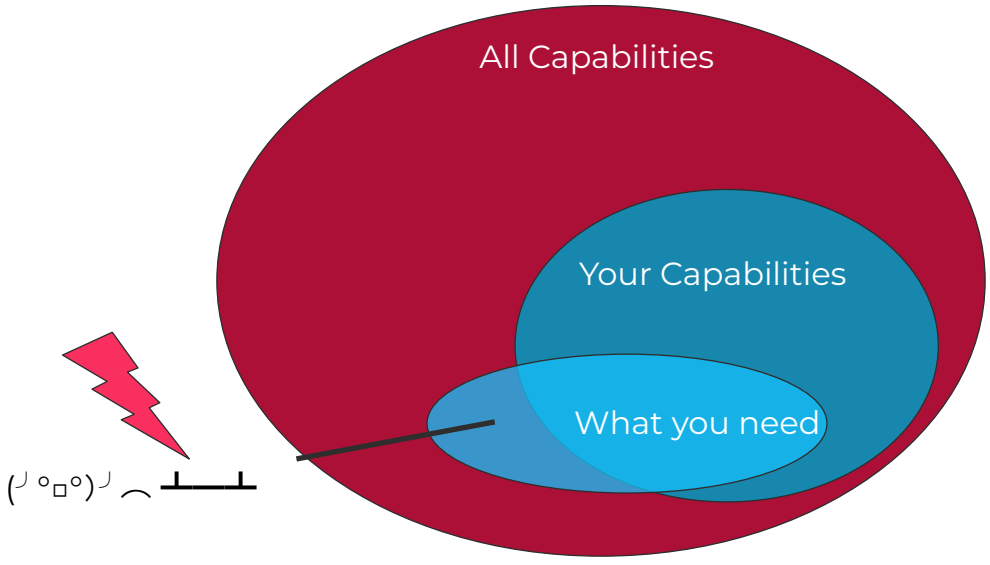
eficode

The **Principle** of **Least Privilege** (PoLP)



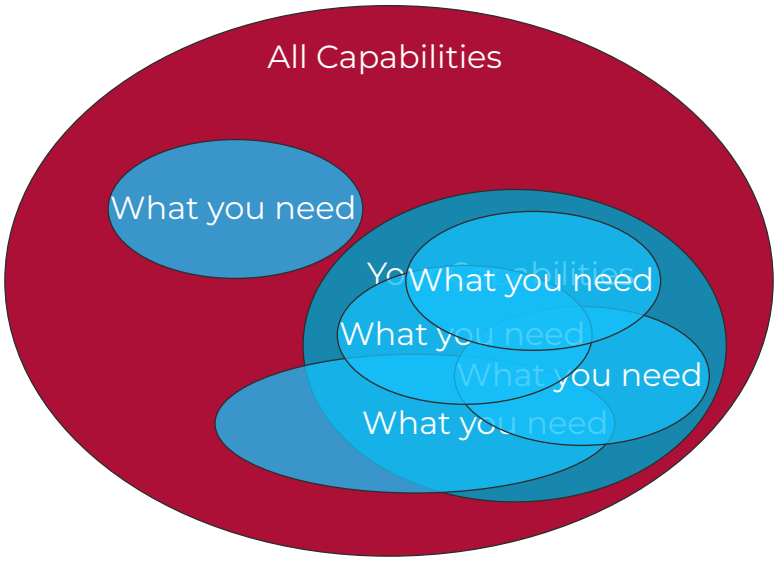
eficode

The **Principle** of **Least Privilege** (PoLP)



eficode

The **Principle** of **Least Privilege** (PoLP)

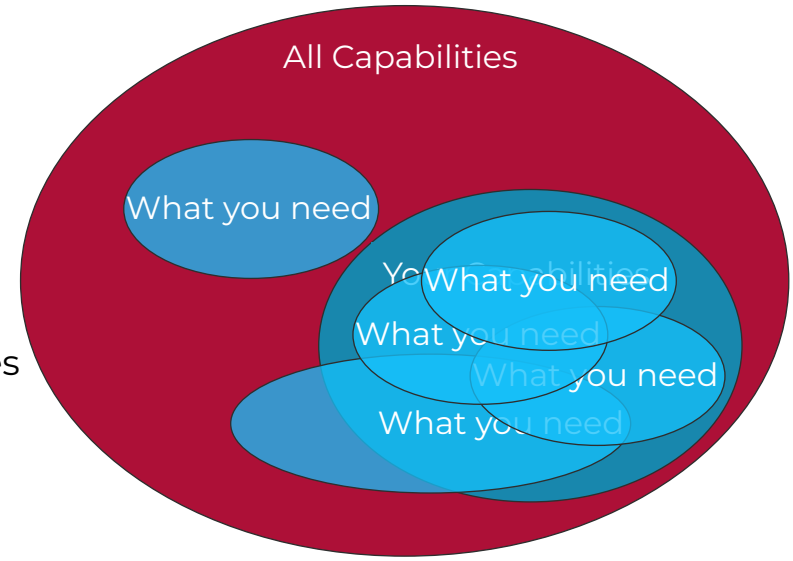


eficode

The **Principle** of **Least Privilege** (PoLP)

Must be **easy** to

1. Switch Capabilities
2. Change Capabilities



eficode

Threat	Violated Property
...	...
...	...

Threat Modeling

“Is this secure?”

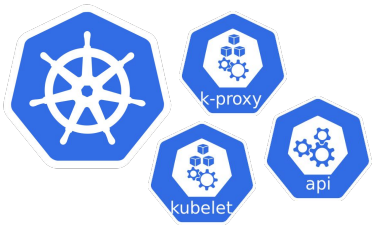
eficode

Threat Modeling

?

eficode

Threat Modeling - Attack Surface



eficode

Threat Modeling - Adversaries

SAMSUNG

SONY

LG

eficode

Threat Modeling - Models - STRIDE

STRIDE: “mnemonic for threats.”

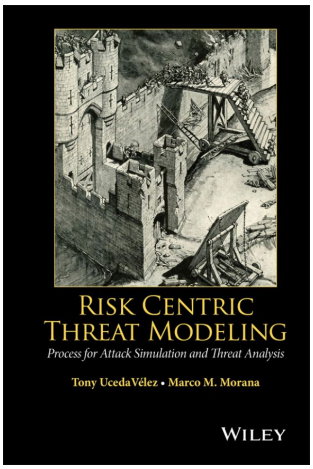
Threat	Violated Property
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-repudiation
Information Disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

eficode

Threat Modeling - Models - PASTA

Process for **A**ttack **S**imulation and **T**hreat **A**nalysis

- Stage 1: Define Objectives
- Stage 2: Define Technical Scope
- Stage 3: Application Decomposition
- Stage 4: Threat Analysis
- Stage 5: Vulnerability & Weakness Analysis
- Stage 6: Attack Modeling
- Stage 7: Risk Mitigation / Countermeasures



PASTA Threat Modeling for Cybersecurity | OWASP All Chapters 2020 Presentation
https://www.youtube.com/watch?v=8k-I3vn8C2A

eficode

Threat Modeling - Models - Persona non Grata example

- Name: John
- Special skills: Kubernetes
- Role: Infrastructure
- Motivation: Angry about salary
- Possible attacks: Cryptojacking
- Remediation: Runtime scanning

https://sysdig.com/blog/detecting-cryptojacking-with-sysdigs-falco/

eficode

Threat Modeling - Models

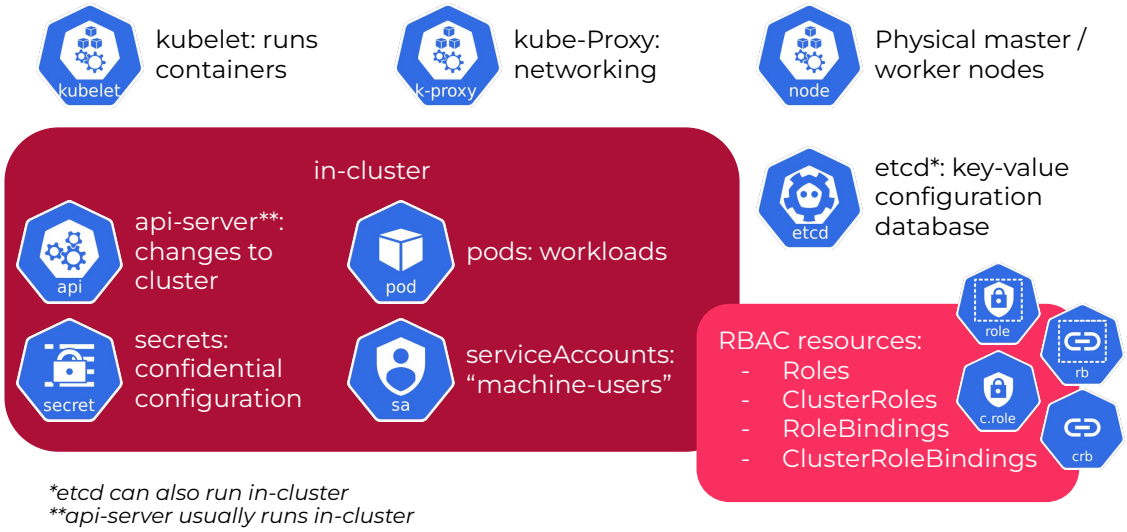
- STRIDE - Centered on “Types of Attacks” or “Threats”
- PASTA - Risk-centric (business impact)
- Persona non grata (“unwanted person”) - ~”Evil User Stories”

... LINDDUN, Security Cards, Trike, CVSS, hTMM, VAST Modeling, Attack Trees, Quantitative TMM, OCTAVE

See: https://insights.sei.cmu.edu/blog/threat-modeling-12-available-methods/

eficode

Threat Modeling - Kubernetes' Attack Surface



*etcd can also run in-cluster
**api-server usually runs in-cluster

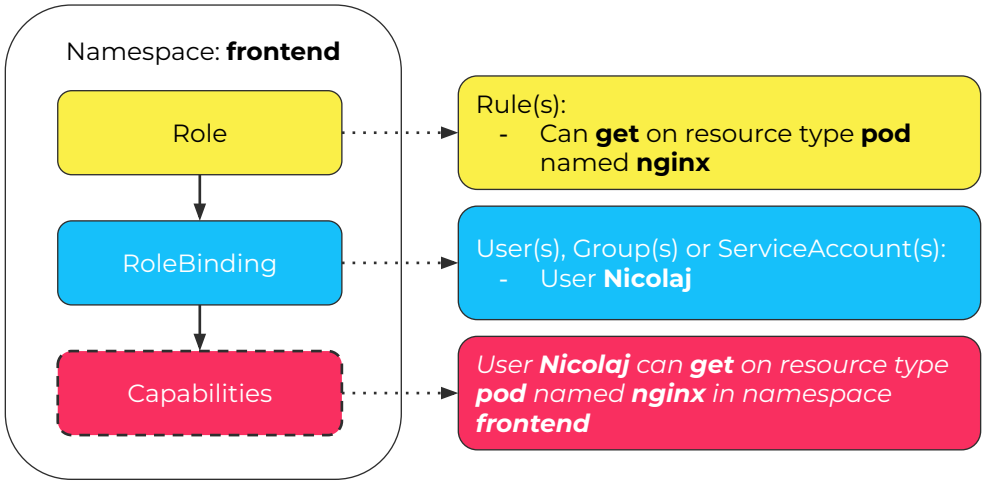
eficode

Role-Based Access Control (RBAC)

Resources in K8s

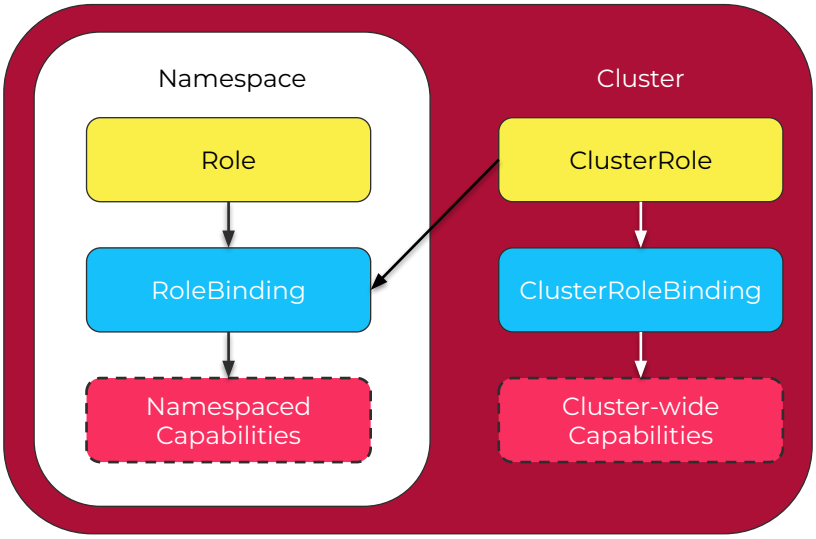
eficode

Role-based Access Control



eficode

Role-based Access Control



Role-based Access Control - Demo

Will Show:

- Getting Access to a K8s cluster

`minikube-rbac-demo.md`

eficode

Impersonation

For test and production!

eficode

Impersonation

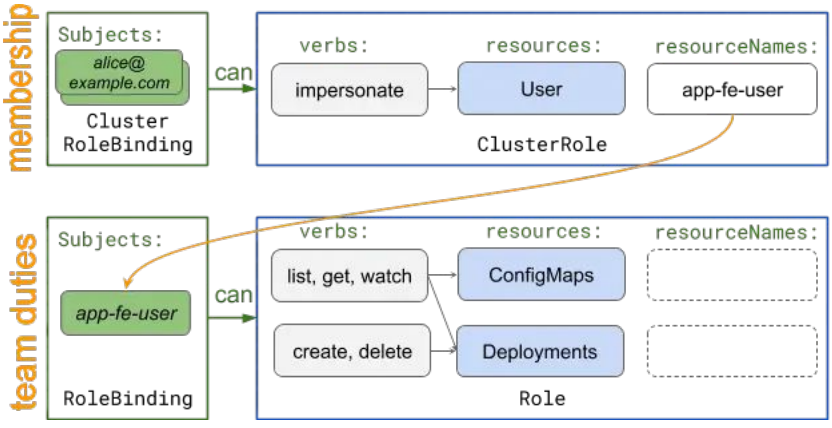
```
kubectl get nodes --as=<username> [--as-group=<group>]
```

- Handy for testing out permissions!
- *Handy for changing permissions!*

https://docs.bitnami.com/tutorials/simplify-kubernetes-resource-access-rbac-impersonation/ - Juanjo Ciarlante, 2 years ago..

eficode

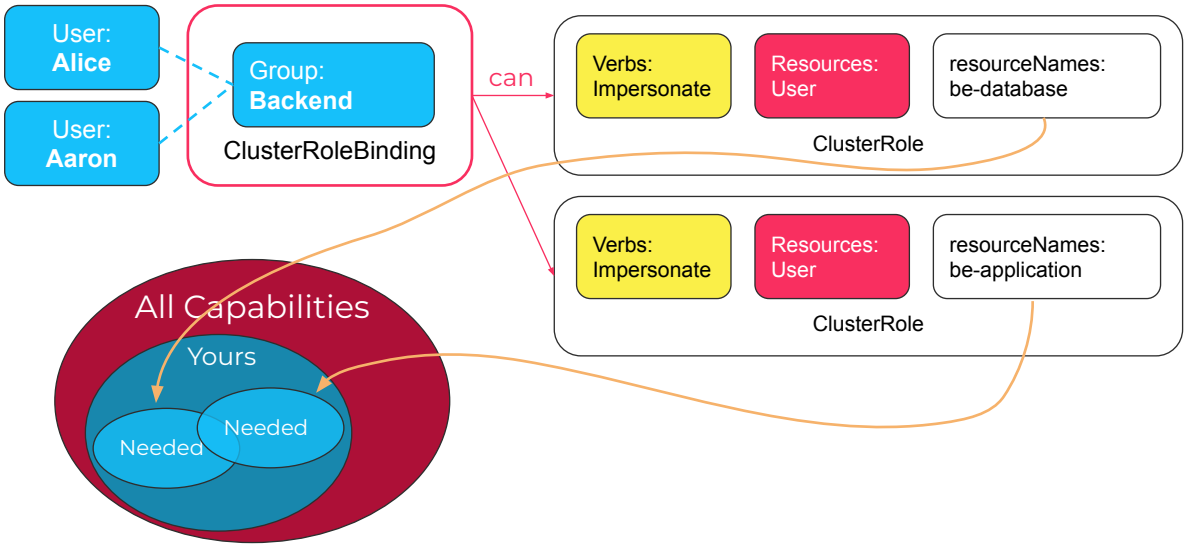
Impersonating - “Virtual Users”



https://docs.bitnami.com/tutorials/simplify-kubernetes-resource-access-rbac-impersonation/
- Juanjo Ciarlante

eficode

Impersonating - “Virtual Users”



eficode

Role-based Access Control - Demo

Will Show:

- Principle of Least Privilege

`minikube-impersonate-polp.md`

eficode

Impersonating - “Virtual Users”

Using Kubectl

```
Kubectl get pods --as=be-database
```

```
Kubectl get pods --as=be-application
```

Using Context, with kubeconfig:

```
kubectl use-context <...>
kubectl get pods
```

```
# kubeconfig
users:
- name: alice/be-application
  user:
    as: be-application
    client-certificate-data: ...
    client-key-data: ...
```

eficode

Impersonation Caveats 

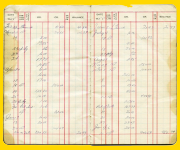
```
... --as=<username> [--as-group=<group>]
```

Using “virtual users” as “groups” feels dirty, but..

- Must set **--as** and for simplicity we just use that as “group.”
- Could hardcode **--as** user to kubeconfig, but then it fails when we don’t add **--as-group**.
- For now we just use **virtual users as groups***

**We’ll still handle original user-groups from LDAP / OIDC / IAM / etc.*

eficode



Audit Logs

Who did what, where and when?

eficode

Audit Logs - basic policy

```
users:apiVersion: audit.k8s.io/v1
kind: Policy
rules:
- level: Metadata
```

Example from: https://minikube.sigs.k8s.io/docs/tutorials/audit-policy/

Levels: **none**, **metadata**, **request**, **requestResponse**

Stored internally (file) or externally (log db)

eficode

Audit Logs - Demo

Will Show:

- Impersonation in Logs

minikube-audit-demo.md

eficode

Audit Logs

```
kubectl get pods --as=alice --as-group=be-database
```

```
{
    ...
    "requestURI":"/api/v1/pods?limit=500",
    "verb":"list",
    "user":{
        "username":"minikube-user",
        "groups":[
            "system:masters",
            "System:authenticated"
        ]
    },
    "impersonatedUser":{
        "username":"alice",
        "groups":[
            "be-database",
            "system:authenticated"
        ]
    }
}
```

eficode

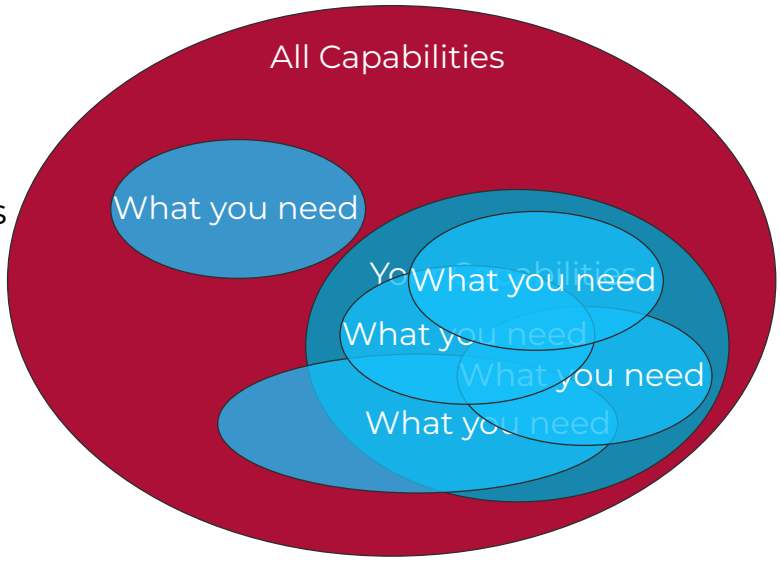
The **Principle** of **Least Privilege** (PoLP) - Revisited

Must be **easy** to

1. Switch Capabilities
2. Change Capabilities

Who elevates privileges?
- **breakglass** virtual user?
- some admin people?

Collaborate as much as possible.
People like to be included.



eficode

“Alternatives”

Do we need this at all? (Yes-ish, we do..)

eficode

Alternatives - kubectl-sudo

- A “wrapper” around Impersonation
- adds **`--as=$USER --as-group system:masters`** to command

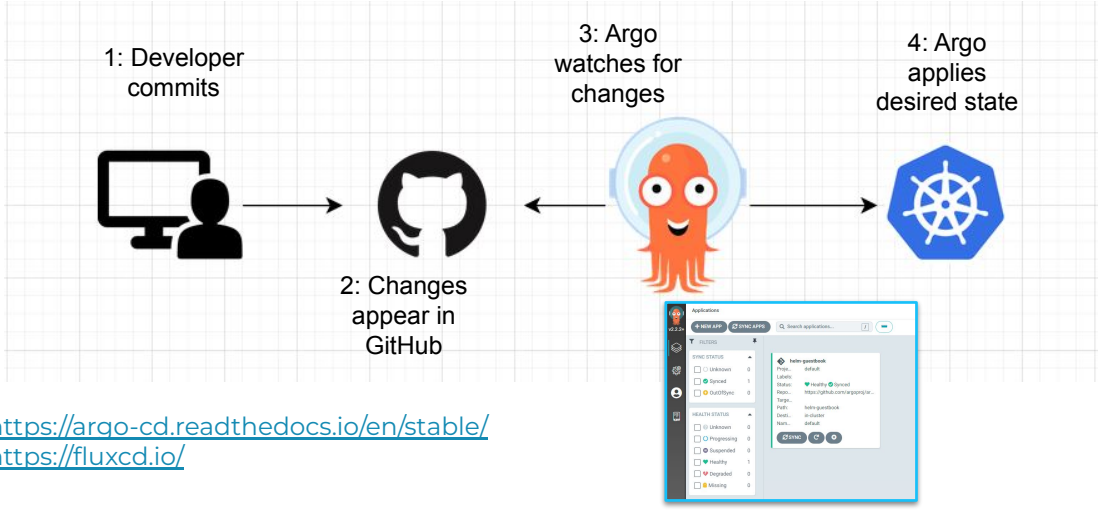
```
$ kubectl get nodes
Error from server (Forbidden): nodes is forbidden: User "bofh" cannot list nodes at the cluster scope

$ kubectl sudo get nodes
NAME                  STATUS   ROLES    AGE   VERSION
kubelet1.example.com  Ready    <none>   96d   v1.11.2
kubelet2.example.com  Ready    <none>   96d   v1.11.2
```

https://github.com/postfinance/kubectl-sudo

eficode

Alternatives - GitOps - ArgoCD (..and Flux)

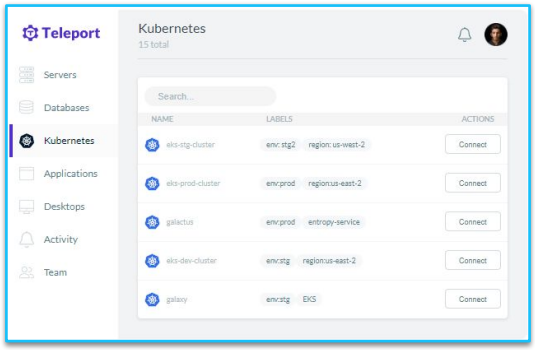


https://argo-cd.readthedocs.io/en/stable/
https://fluxcd.io/

eficode

Alternatives - Teleport

- Manage access to Clusters, Databases, Desktops, etc.
- Proxy interface for interacting with kubectl
- Replay sessions from the UI

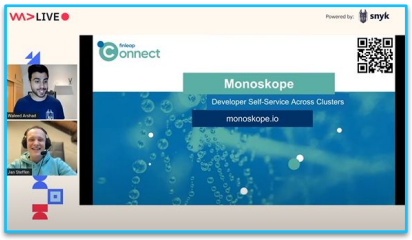


https://goteleport.com/

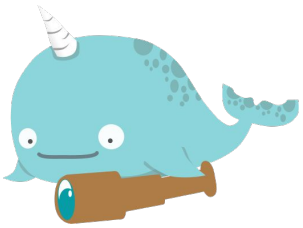
eficode

Alternatives - Monoskope

WeAreDevelopers Live - DevOps Day, 2021
https://youtu.be/8FPXqjwSMdI?t=7382



- Self-service multi-cluster
- Operator + CLI (M8)
- Event Sourcing, auditing as a first-class citizen
- Reactors change state depending on requests
- Requests are made by Users
- Connects to OIDC and assigns scopes to users

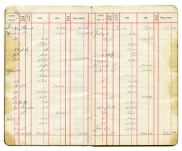


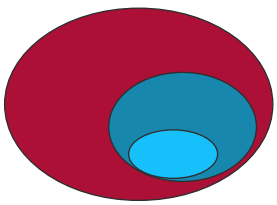
https://monoskope.io/

eficode

Alternatives - Viability

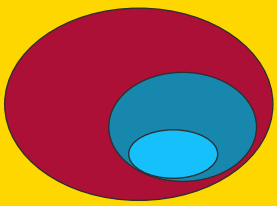
- Won't get rid of RBAC; still shouldn't use default **ServiceAccounts** for workloads.
- GitOps is Awesome!
- RBAC is Awesome!





Threat	Violated Property
...	...
...	...

eficode



Thank you!

.. questions? 😃

Threat	Violated Property
...	...
...	...