

# Don't talk to strangers

Test isolation with containers

# Hristo Iliev

- SAP
  - NetWeaver Application Server Java
  - SAP HANA Cloud Platform
- Open Source
  - Eclipse Virgo
  - Cloud Foundry (Diego & Abacus)



# Georgi Sabev

- SAP
  - SAP HANA Cloud Platform
- Open Source
  - Cloud Foundry (Diego & Abacus)



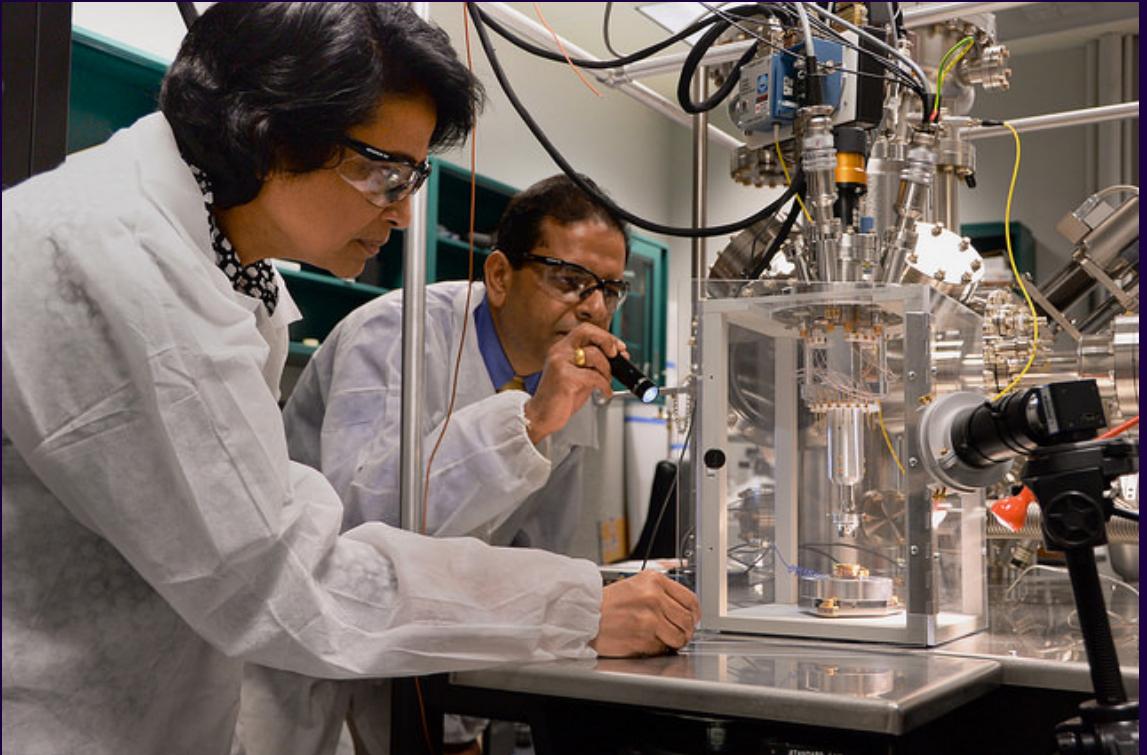
A close-up photograph of the side and rear of a silver car, possibly a hatchback, parked on a dark, rocky surface. In the background, a small white boat with a blue stripe is visible on the water under a cloudy sky.

# Pictures

Creative Commons @ Flickr

# Creating Working Software

- Tests
- Automation
- Discipline





# Problems

# Test Pollution

- Environment variables
- Database records
- Files
- Programs / libraries



# Resource contention

- Processes compete for
  - CPU
  - Memory
  - I/O



# Multiple environments

- OS
  - Mobile / Desktop
  - Cloud / Standalone
  - Databases
  - Versions



# Test problems summary

- Test pollution
- Resource contention
- Multiple environments



# Solutions

- Traditional
  - Dedicated hardware
  - Virtual machines
  - Cloud: AWS, Azure
- New tools

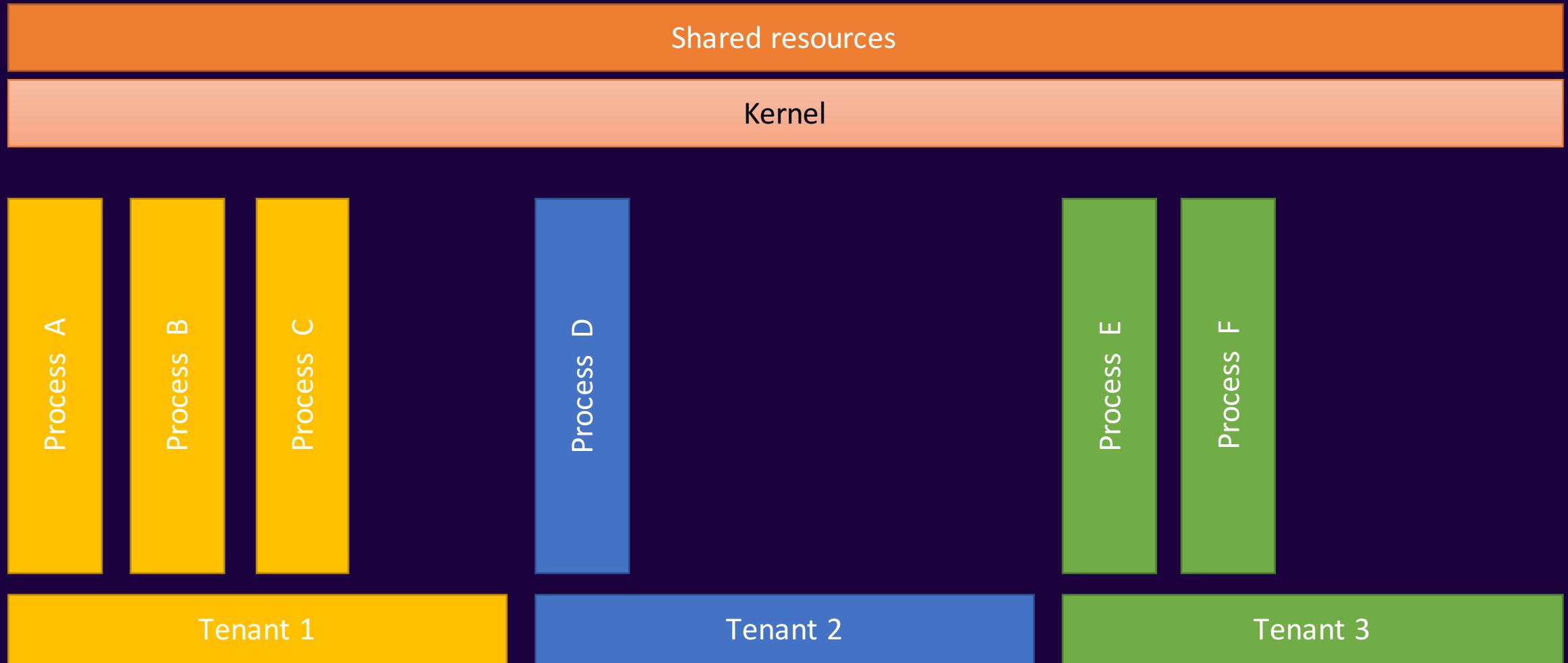




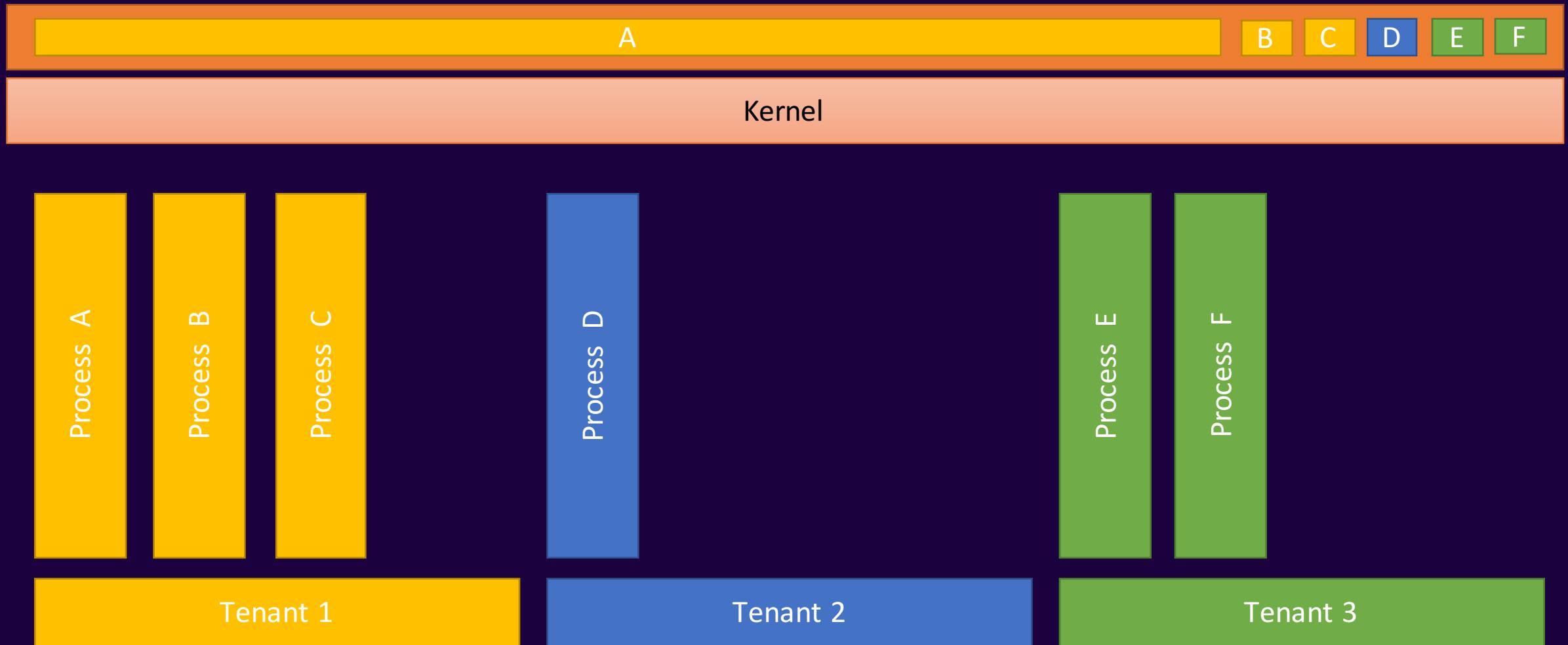
# Containers

build, isolate, ship

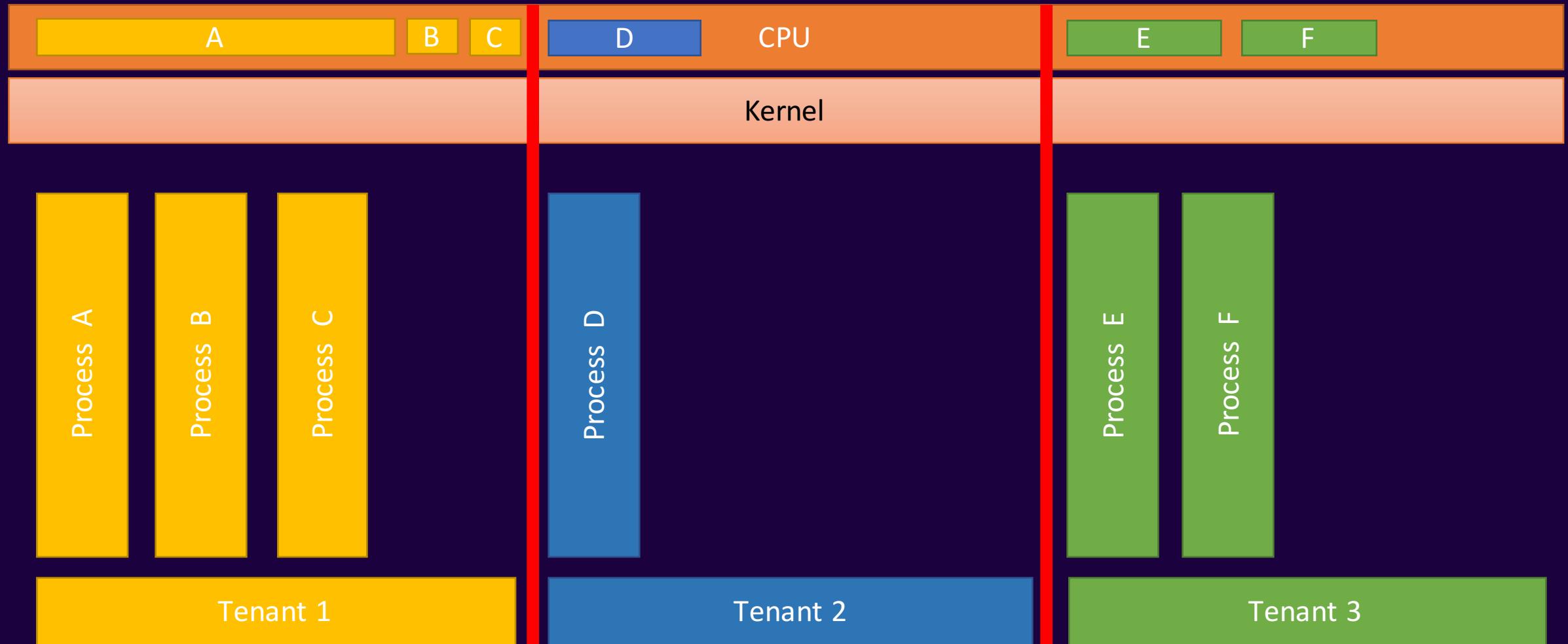
# Container: Resource isolation



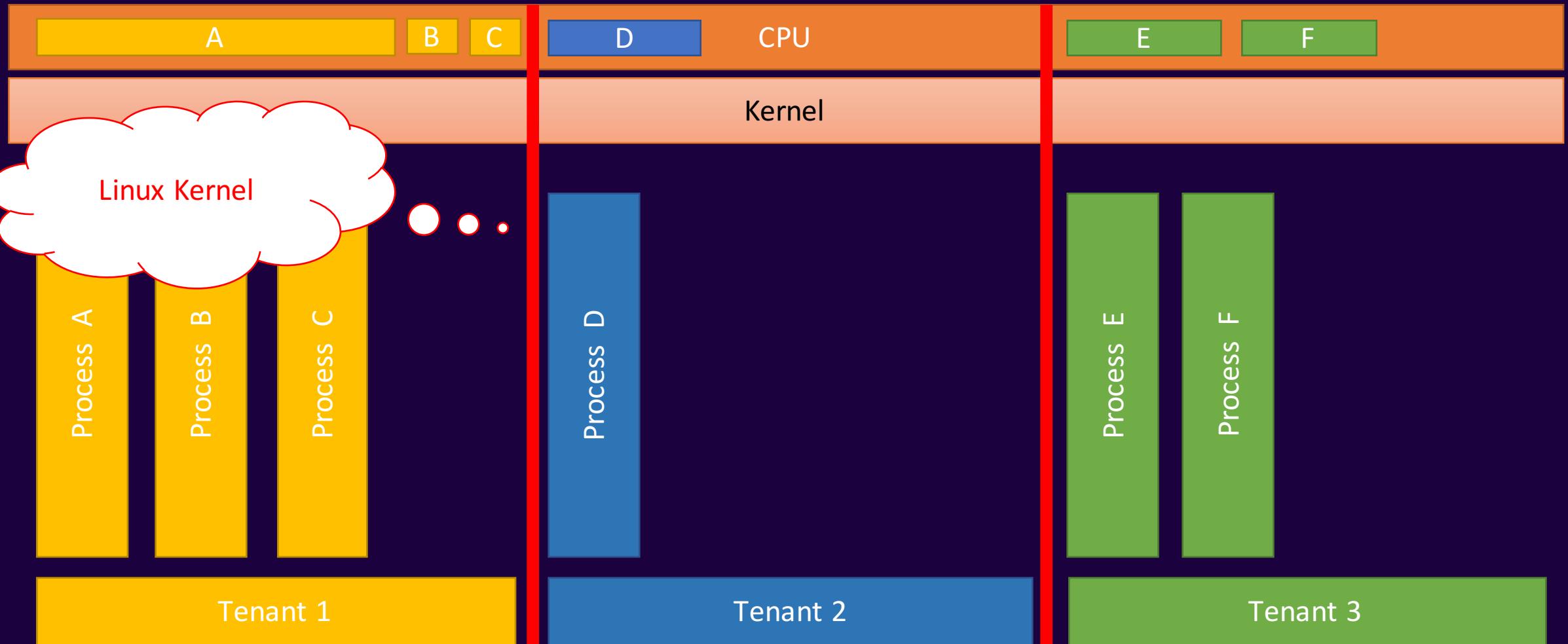
# Container: Resource isolation



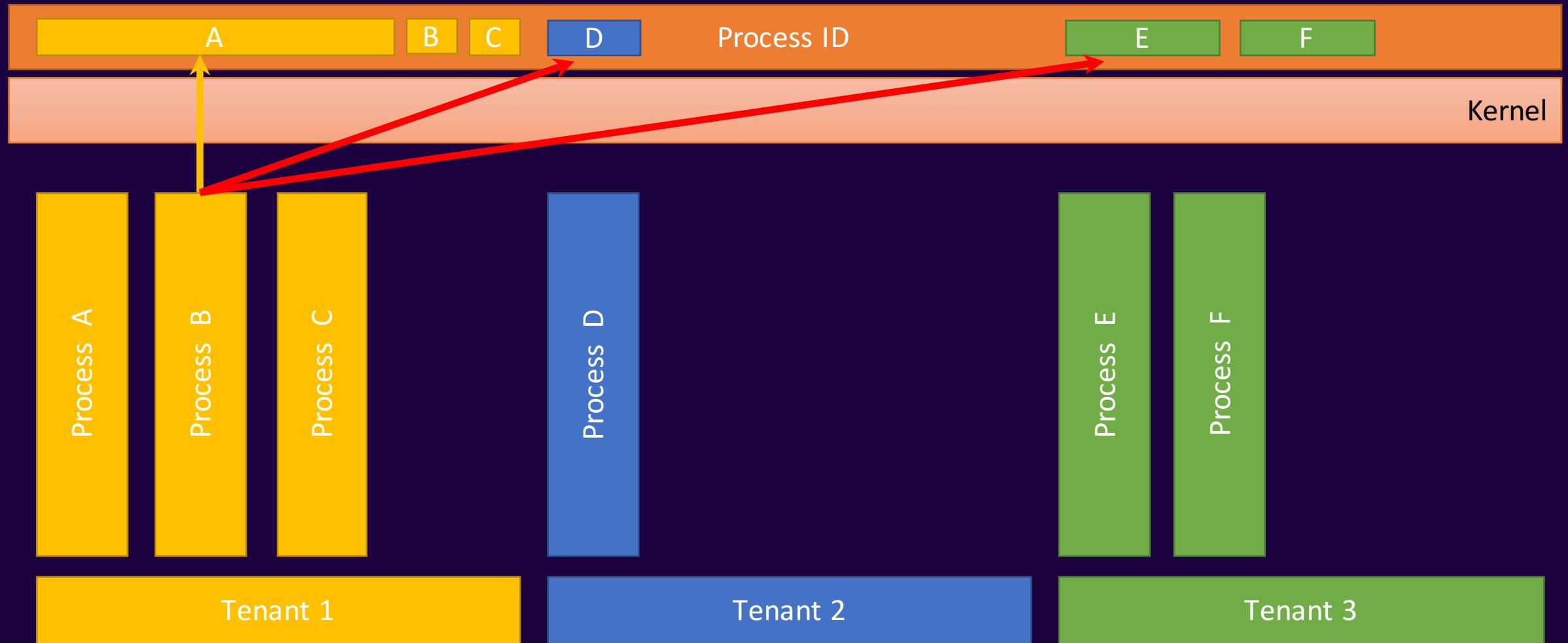
# Container: Resource isolation



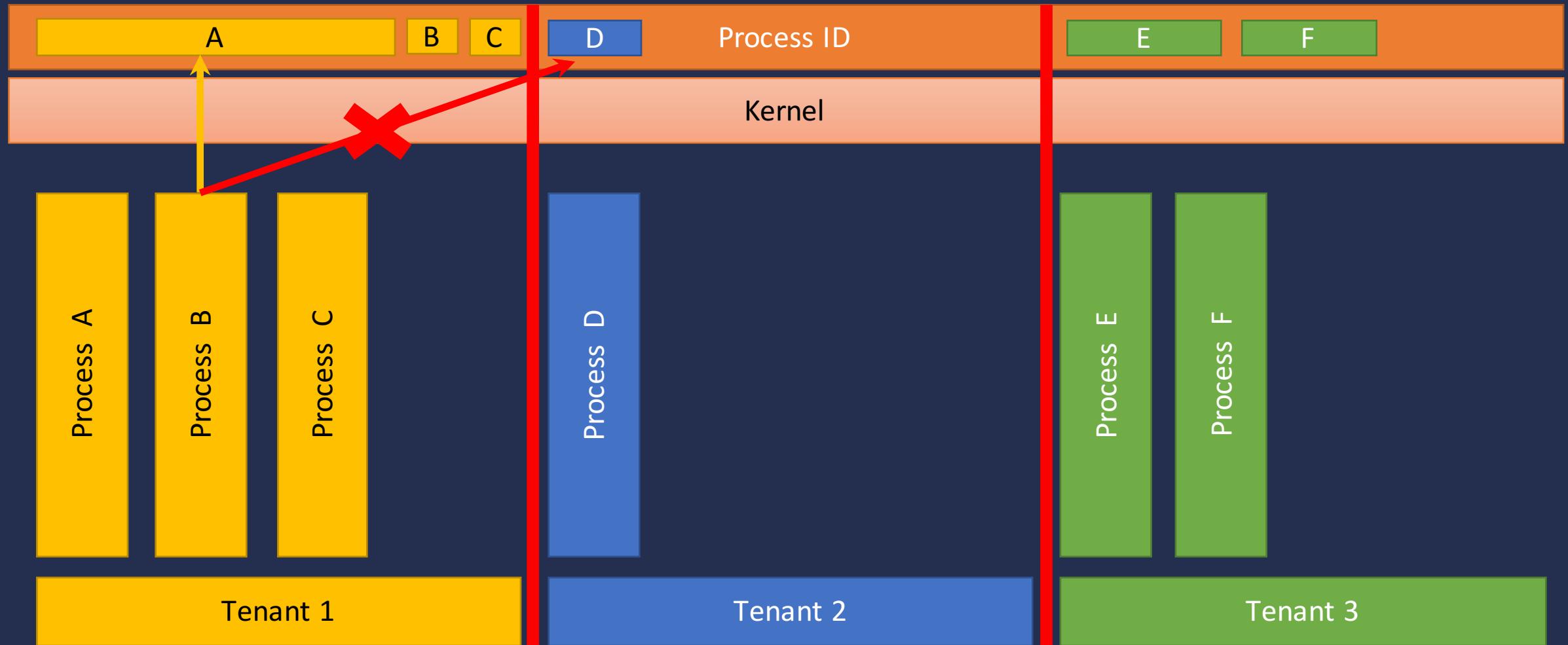
# Container: Resource isolation



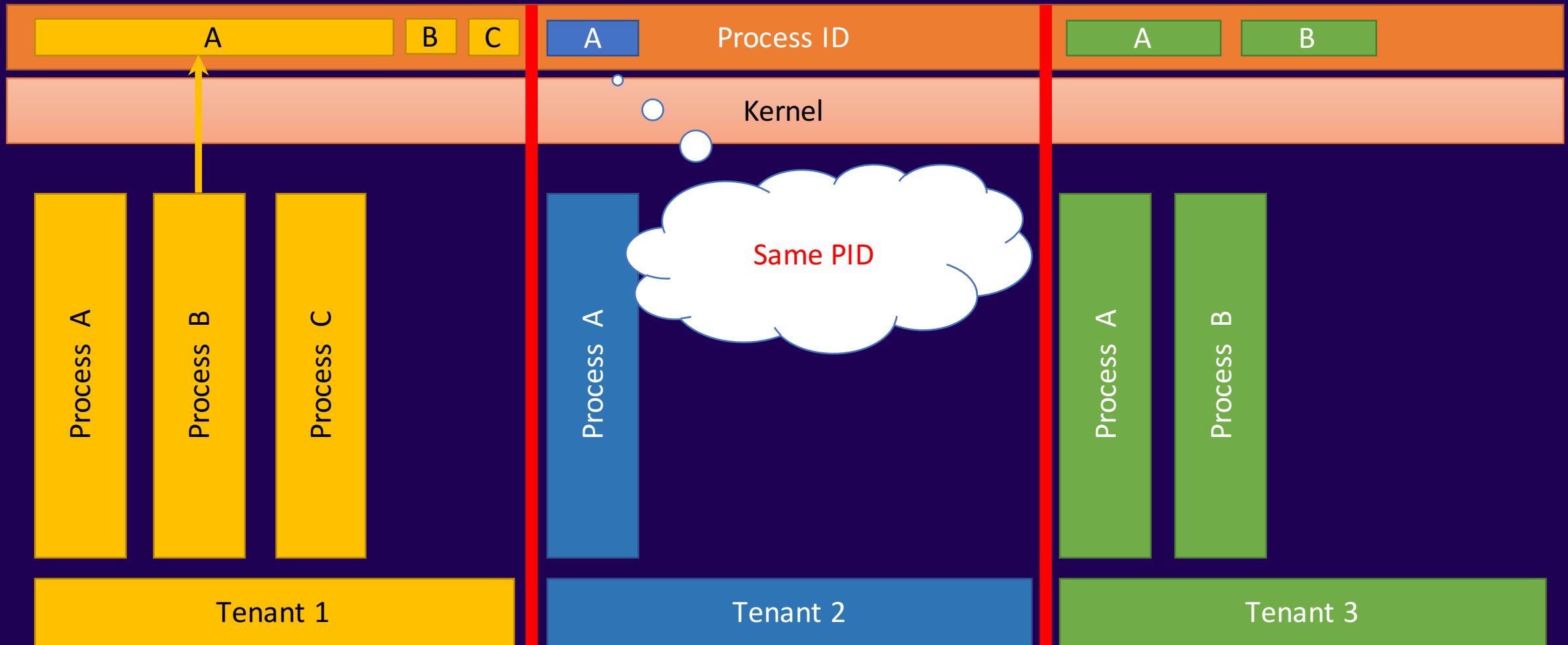
# Container: Namespace isolation



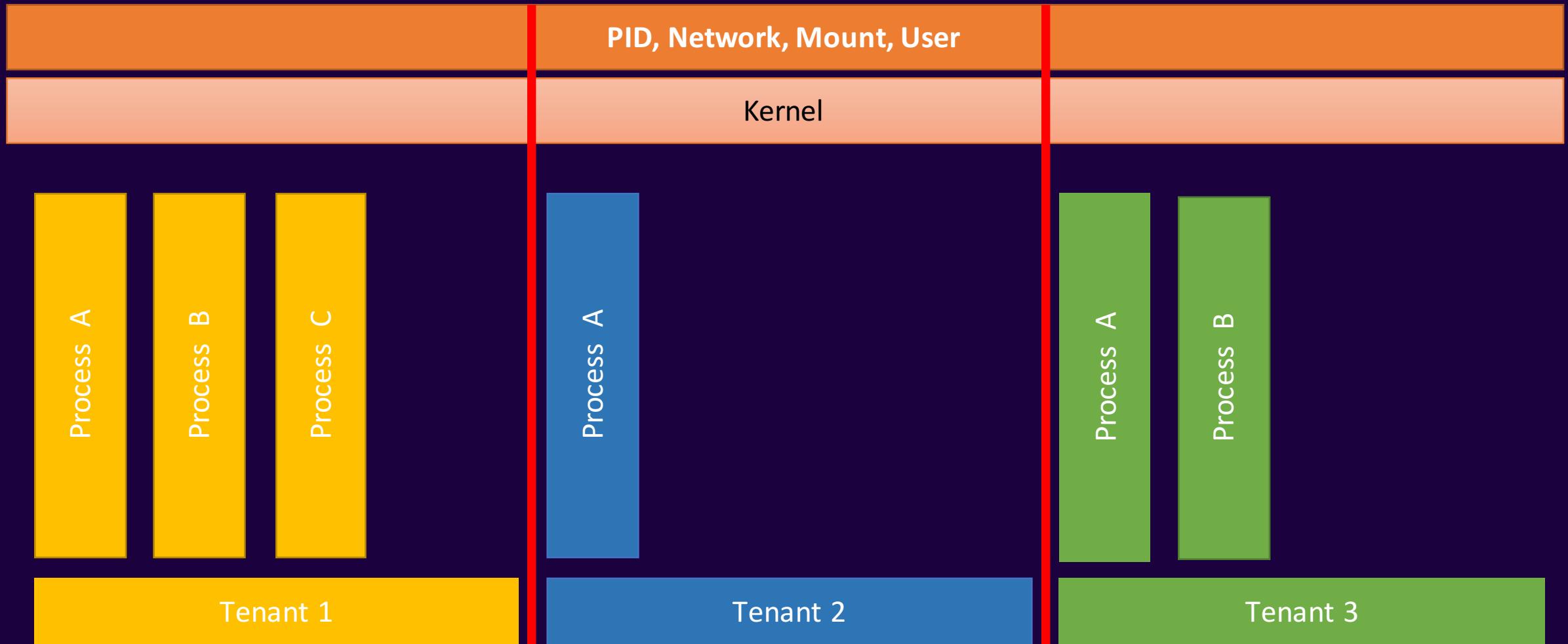
# Container: Namespace isolation



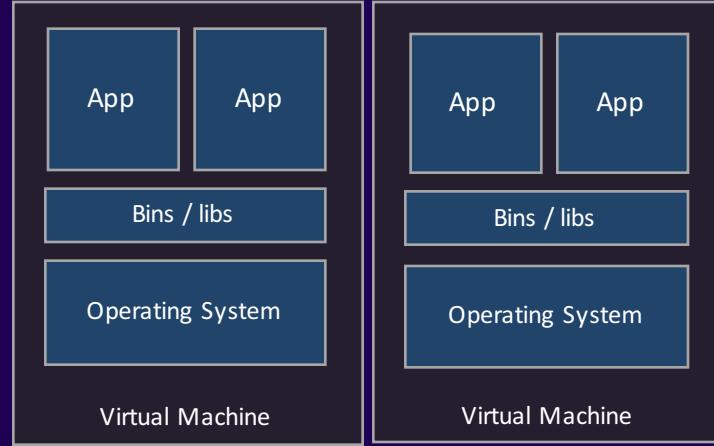
# Container: Namespace isolation



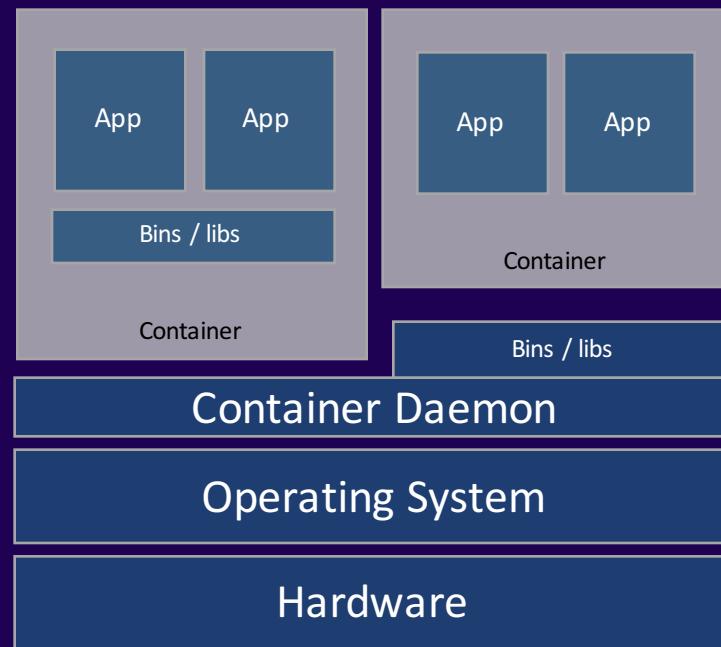
# Container: Namespace isolation



# Containerization = Lightweight Virtualization



**Virtualization**

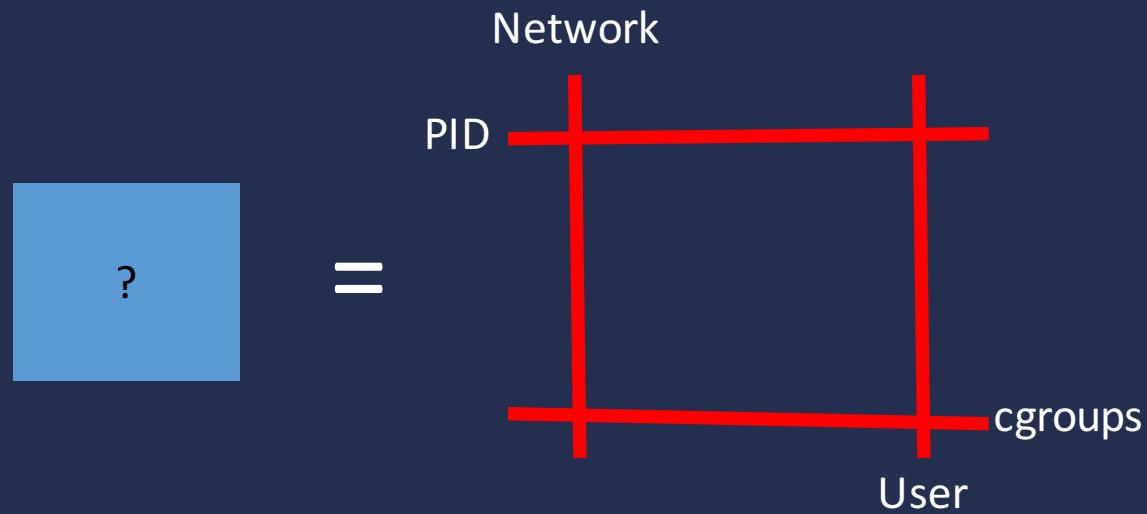


**Containerization**

# Container

?

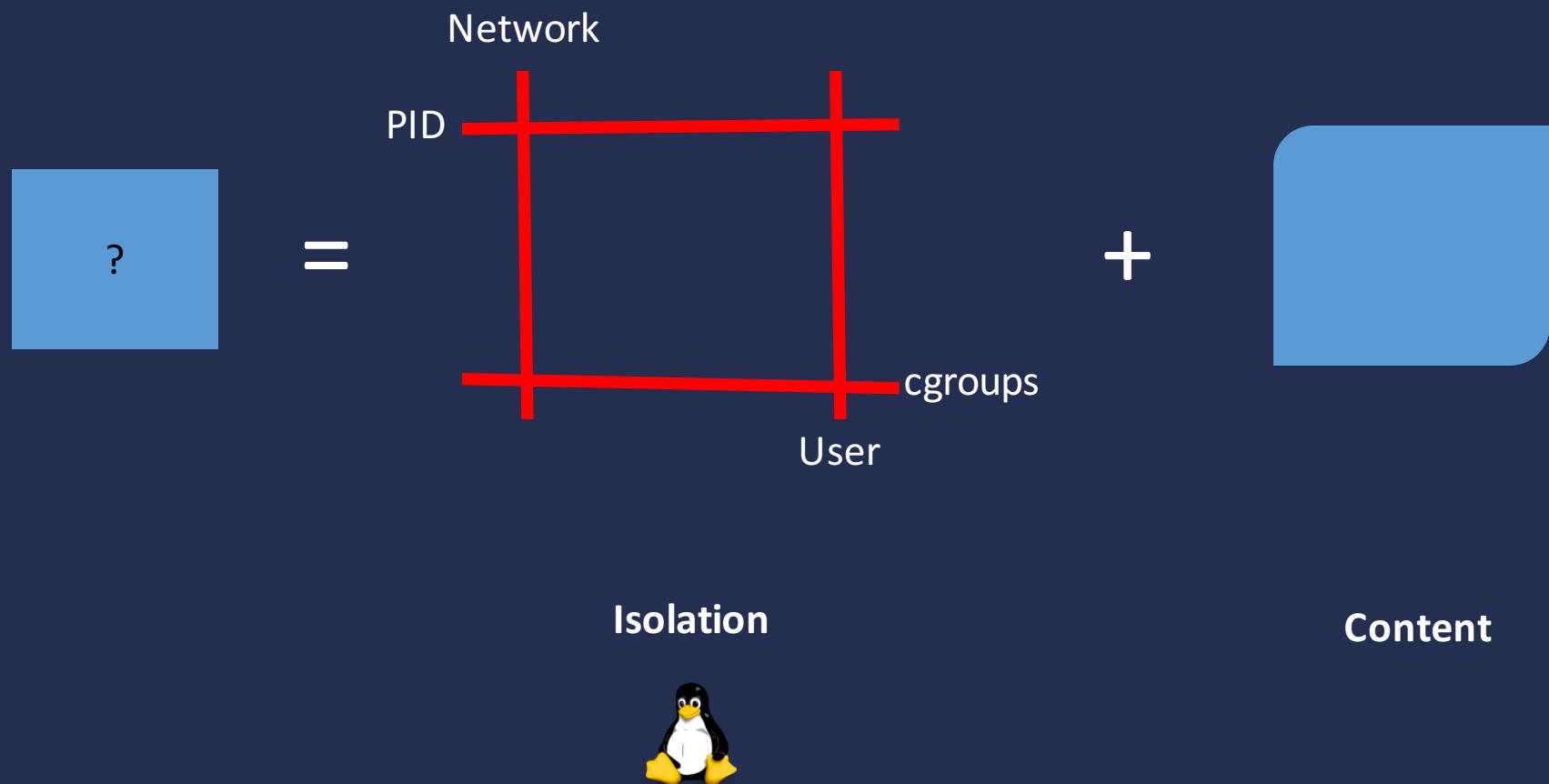
# Container



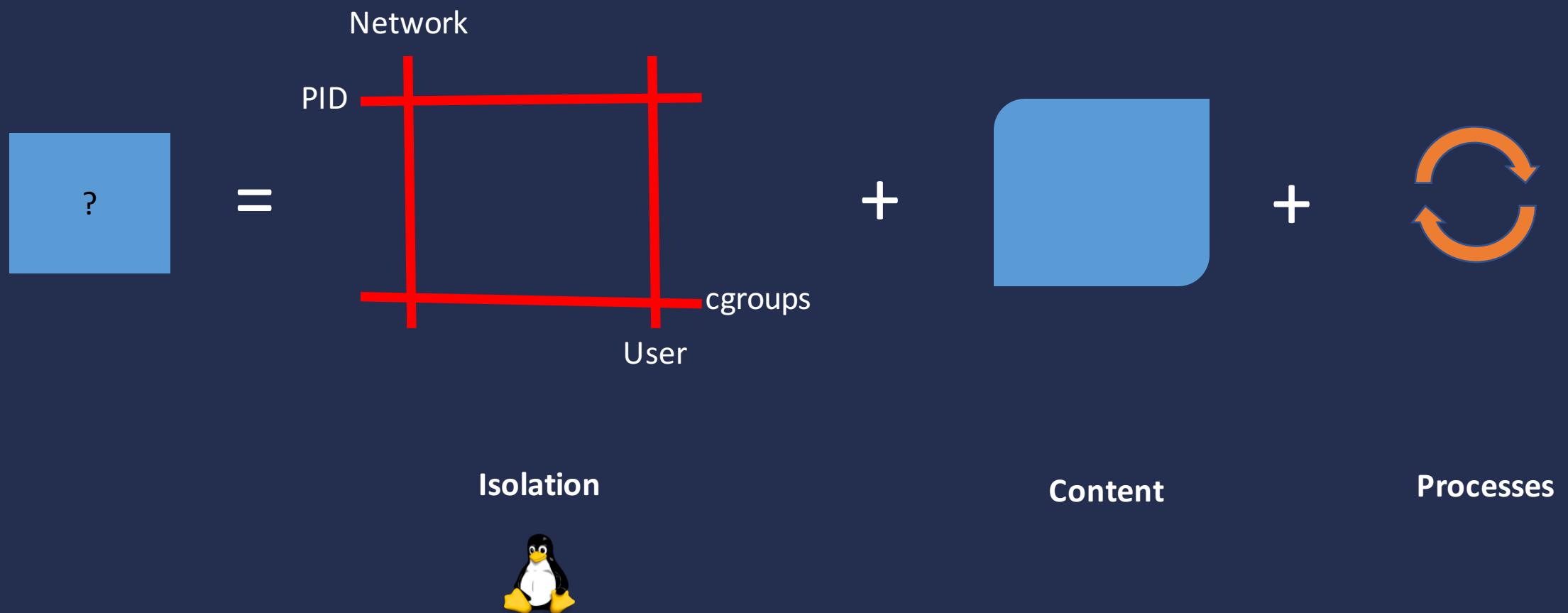
Isolation



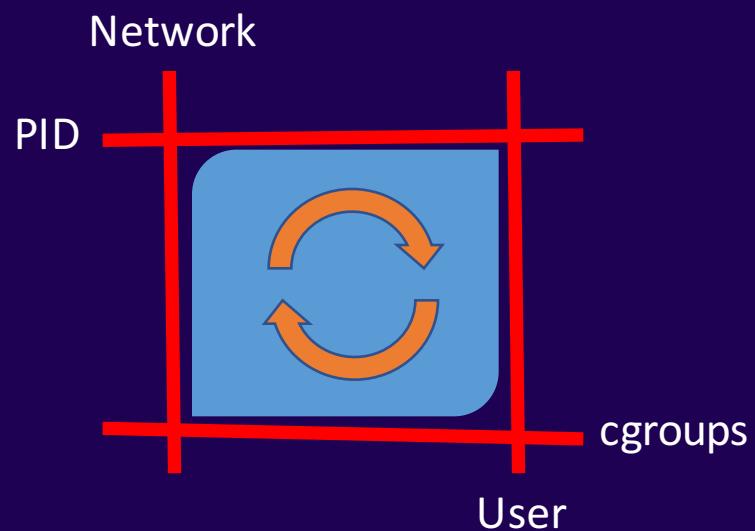
# Container



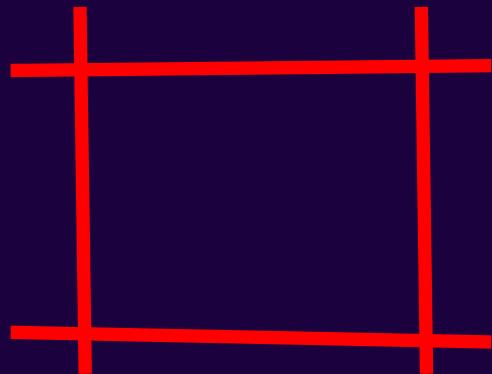
# Container



# Container



# Container actions



**Make me a container**



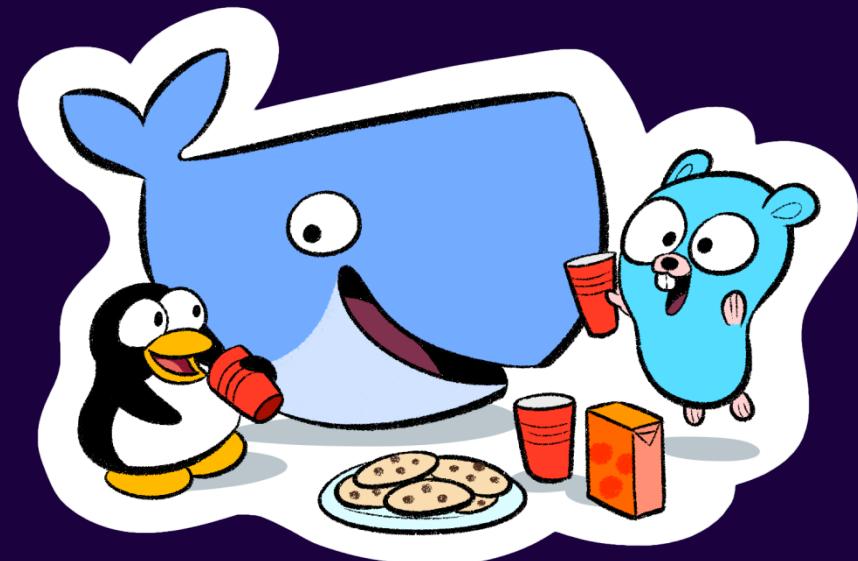
**Put this in it**



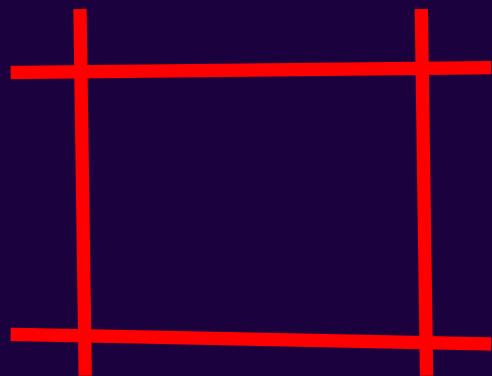
**Run this for me**

# Docker

- “Build, Ship, Run” **Containers**
  - Dockerfile
  - Docker image
  - Docker Hub



# Dockerfile



**Make me a container**



**Put this in it**



**Run this for me**

# Dockerfile

- Example:

```
FROM debian:stable
RUN apt-get update && apt-get install -y --force-yes apache2
EXPOSE 80 443
VOLUME [ "/var/www", "/var/log/apache2", "/etc/apache2" ]
ENTRYPOINT [ "/usr/sbin/apache2ctl", "-D", "FOREGROUND" ]
```

# Dockerfile

- Example:

```
FROM debian:stable
RUN apt-get update && apt-get install -y --force-yes apache2
EXPOSE 80 443
VOLUME [ "/var/www", "/var/log/apache2", "/etc/apache2" ]
ENTRYPOINT [ "/usr/sbin/apache2ctl", "-D", "FOREGROUND" ]
```

# Dockerfile

- Example:

```
FROM debian:stable
RUN apt-get update && apt-get install -y --force-yes apache2
EXPOSE 80 443
VOLUME [ "/var/www", "/var/log/apache2", "/etc/apache2" ]
ENTRYPOINT [ "/usr/sbin/apache2ctl", "-D", "FOREGROUND" ]
```

# Dockerfile

- Example:

```
FROM debian:stable
RUN apt-get update && apt-get install -y --force-yes apache2
EXPOSE 80 443
VOLUME [ "/var/www", "/var/log/apache2", "/etc/apache2" ]
ENTRYPOINT [ "/usr/sbin/apache2ctl", "-D", "FOREGROUND" ]
```



“



<https://github.com/hsiliev/strangers>

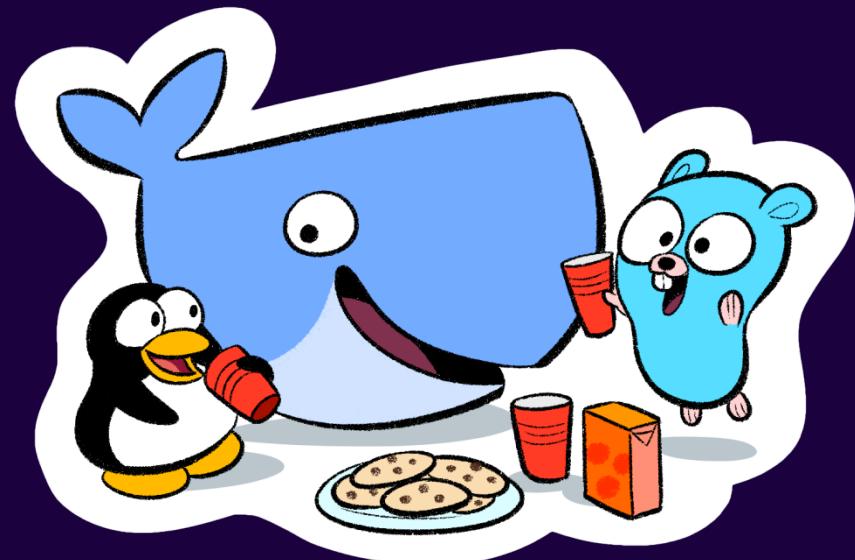
“

# Demo



# Containers: Pros & Cons

- Pros
  - Isolation
  - Reproducible environment
  - Easy to maintain
- Cons
  - Multi-platform
  - Discipline



# Software Production

- Complexity
- Tests
- Automation





# Pipelines

## Multiple-builds

# Popular solutions

- Build servers
  - Jenkins
  - Travis
  - ...
- Homegrown





# Concourse

## CI, Pipelines, Delivery

# Concourse

- CI
  - Original name: “Concorde”
  - Reproducible builds
  - Simple building blocks
  - Docker containers

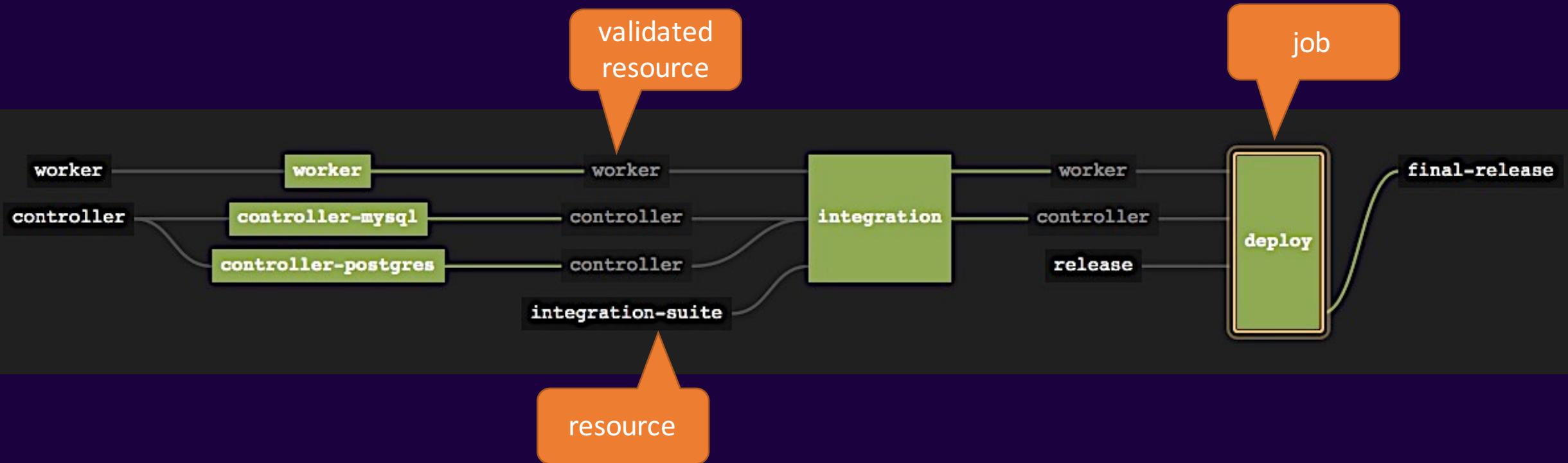


# Concourse: Concepts

- Resources
- Tasks
- Jobs



# Concourse: Pipeline



# Concourse: Pipeline YML

resources:

- name: source

type: git

source:

uri: source.git

branch: master

jobs:

- name: build

plan:

- get: source

- task: build

file: build.yml

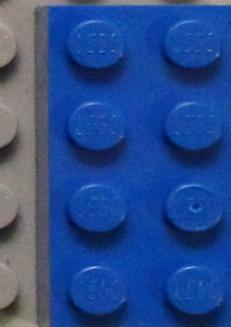
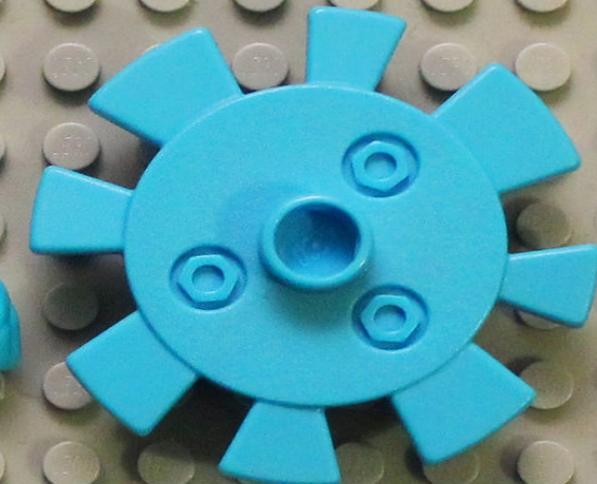
“



<https://github.com/hsiliev/strangers>

“

# Demo



# Concourse: Pros & Cons

- Pros
  - Reproducible environment
  - Native pipeline support
  - Release process modeling
- Cons
  - Maturity
  - Pipeline management
  - Scripting



# Thank you!



[hristo.iliev@sap.com](mailto:hristo.iliev@sap.com)



[georgi.sabev@sap.com](mailto:georgi.sabev@sap.com)