How Migrating Jenkins to Kubernetes Broke our Brains

github.com/solarhess/jenkins_kube_brains

Jonathan Hess & Loren Mountain Trout, SAP
December, 2018

Welcome

Agenda

Bad old Jenkins

Jenkins on K8s in just 3 easy steps

What happens when you let the developers drive

You can do this too

Questions


The story of our
bad old Jenkins.

Requirements for new Jenkins

Simple enough that a developer can manage it

On the corporate network (not the cloud)

Free

Efficient and Repeatable

Fast

3 easy steps to a new Jenkins

Prepare VMs

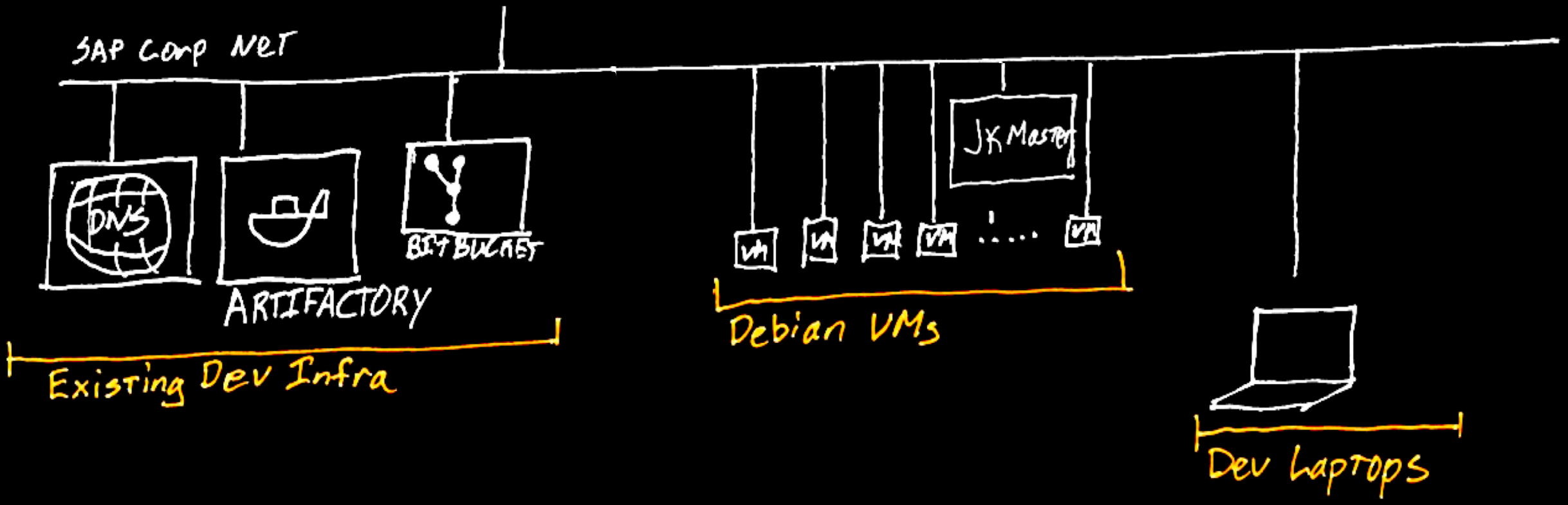
Prepare Kubernetes

Configure Jenkins

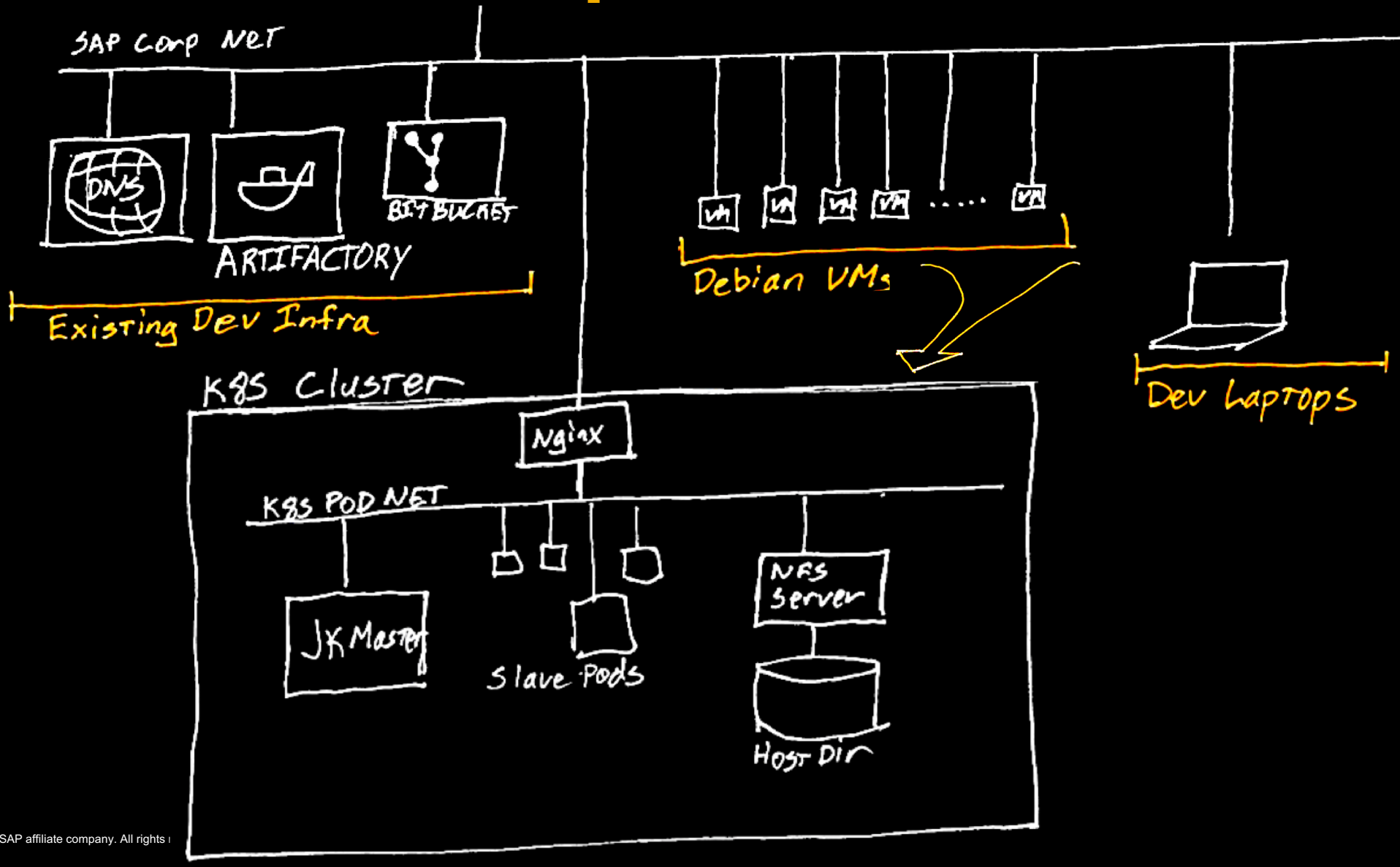

“Kubernetes is
open source”

github.com/solarhess/jenkins_kube_brains

What we started with



What we ended up with



Kubernetes broke our brains

Prepare VMs

Why is it slow and hanging?

Prepare Kubernetes

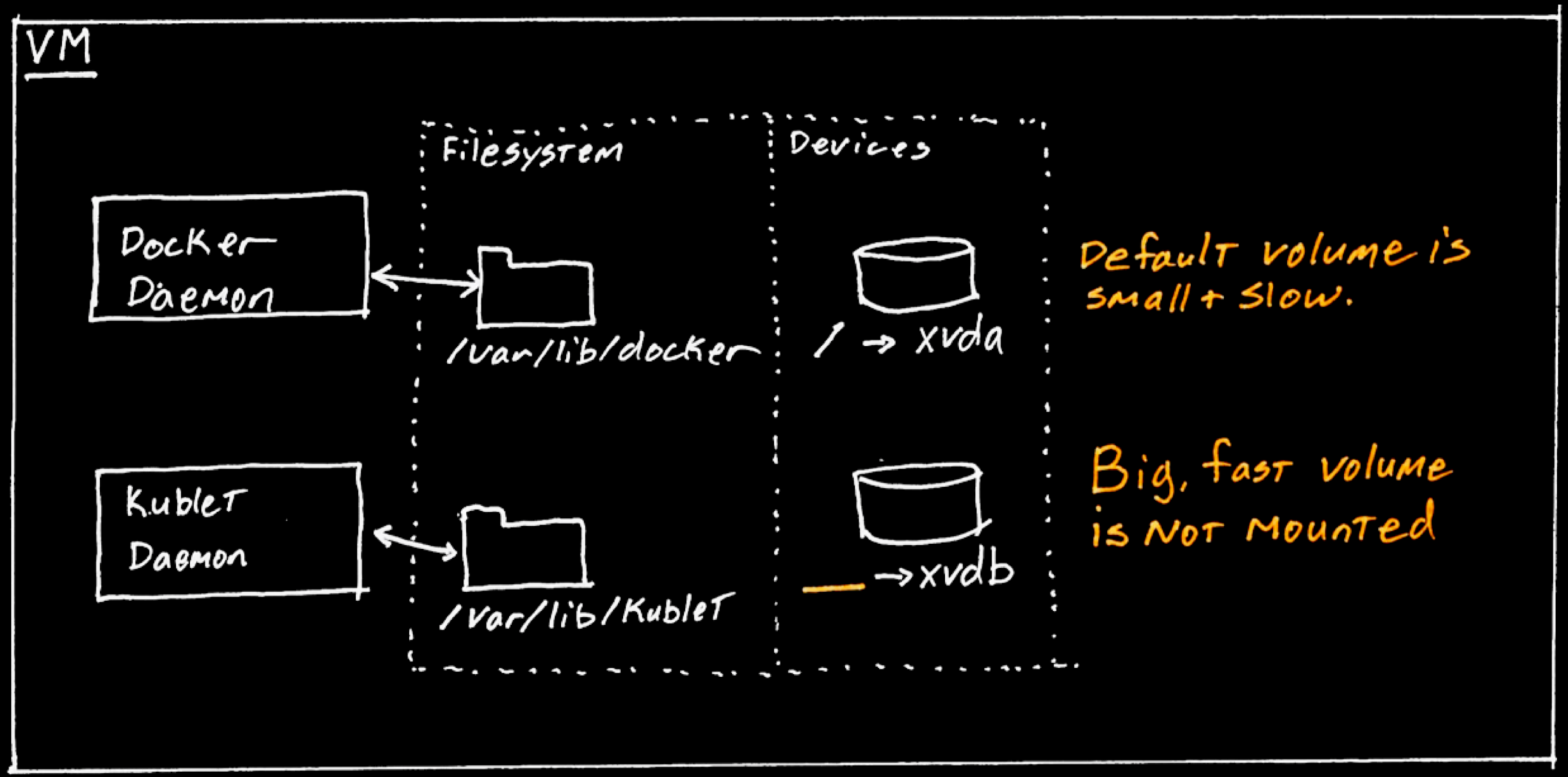
What's up with the network?

Prepare Jenkins

Storage for Jenkins Home?

Prepare VMs: Why is it slow and hanging?

VMs with attached volumes



Mount and move

```
64    file -s /dev/xvdb
65    mkfs -t ext4 /dev/xvdb
66    mkdir /var-alt
67    mount /dev/xvdb /var-alt
```

Don’t forget to update /etc/fstab

```
77    mkdir -p /var-alt/lib/docker
78    cat << EOF > /etc/docker/daemon.json
79    {
80      "exec-opts": ["native.cgroupdriver=cgroupfs"],
81      "graph": "/var-alt/lib/docker"
82    }
83    EOF
84
```

```
89    rm -rf  /var/lib/kubelet
90    rm -rf  /var-alt/lib/kubelet
91
92    mkdir -p /var-alt/lib/kubelet
93    ln -s /var-alt/lib/kubelet /var/lib/kubelet
94    mkdir -p /etc/systemd/system/kubelet.service.d/
95
```

Prepare Kubernetes: What's up with the network not connecting?

IP address ranges

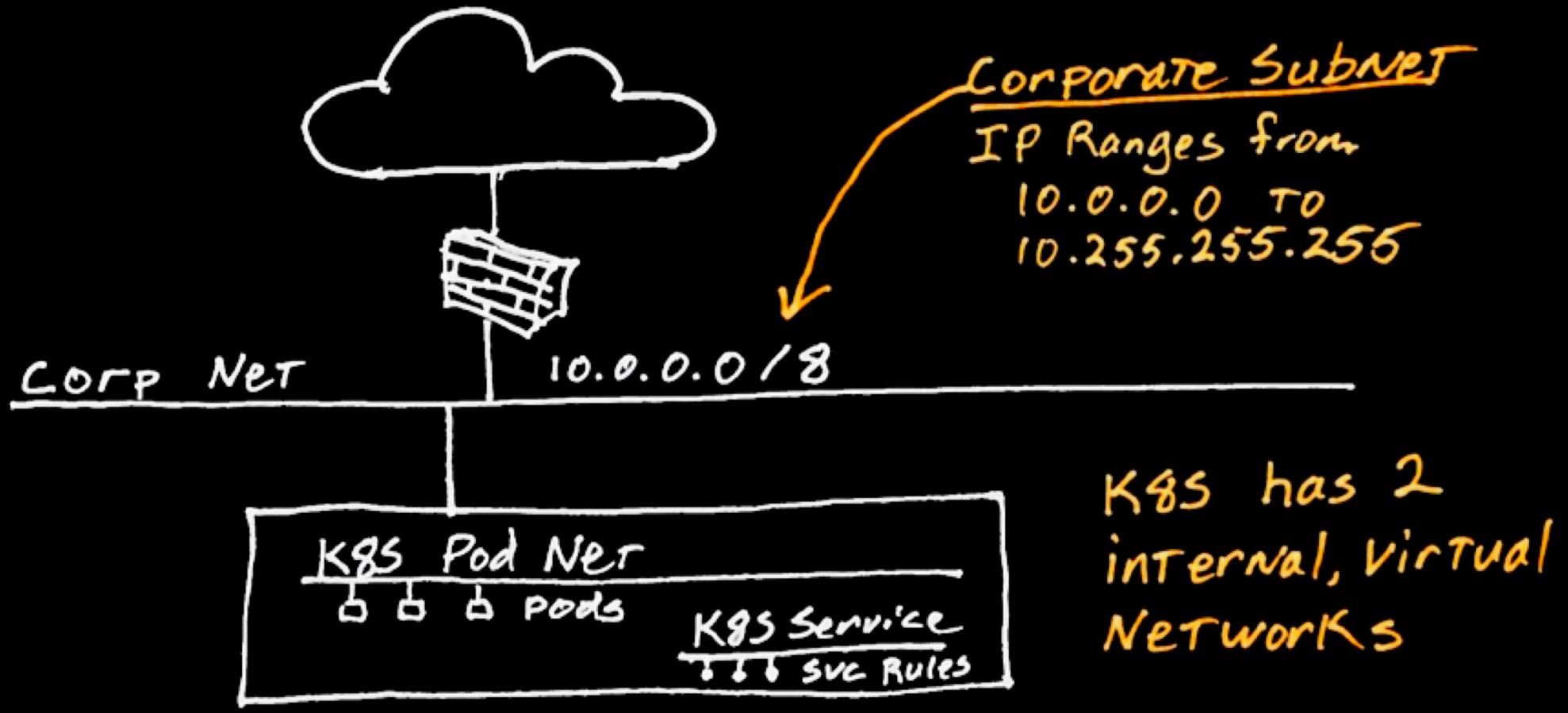
RFC 1918 Address Allocation for Private Internets February 1996

3. Private Address Space

The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:

10.0.0.0	-	10.255.255.255	(10/8 prefix)
172.16.0.0	-	172.31.255.255	(172.16/12 prefix)
192.168.0.0	-	192.168.255.255	(192.168/16 prefix)

An enterprise that decides to use IP addresses out of the address space defined in this document can do so without any coordination with IANA or an Internet registry. The address space can thus be used by many enterprises. Addresses within this private address space will only be unique within the enterprise, or the set of enterprises which choose to cooperate over this space so they may communicate with each other in their own private internet.



kubeadm init

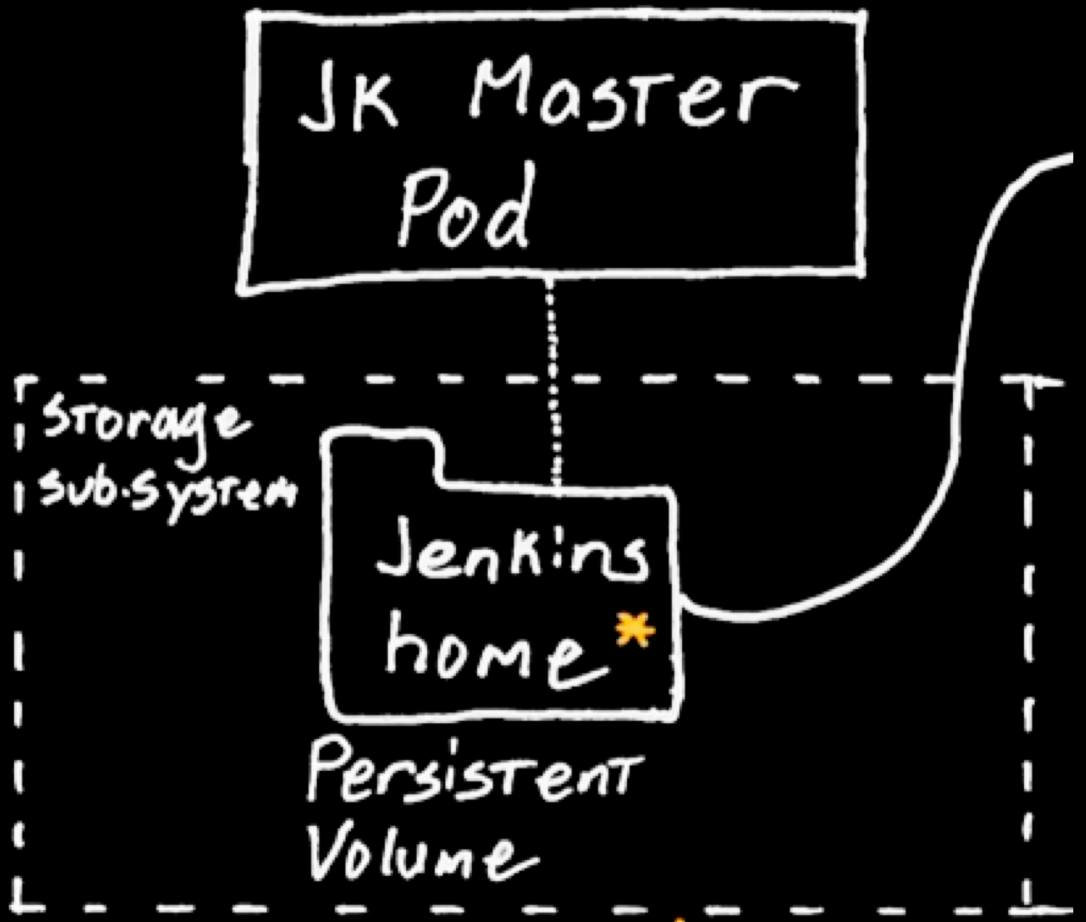
```
34  POD_NETWORK_CIDR=192.168.0.0/17
35  SERVICE_NETWORK_CIDR=192.168.128.0/17
```

```
32  sudo kubeadm init \
33      --ignore-preflight-errors cri \
34      --token-ttl 0 \
35      --pod-network-cidr=${POD_NETWORK_CIDR} \
36      --apiserver-cert-extra-sans=${MASTER_NODE_HOSTNAME} \
37      --service-cidr=${SERVICE_NETWORK_CIDR} | tee kubeadm-i
```

Prepare Jenkins: Where do we put Jenkins Home?

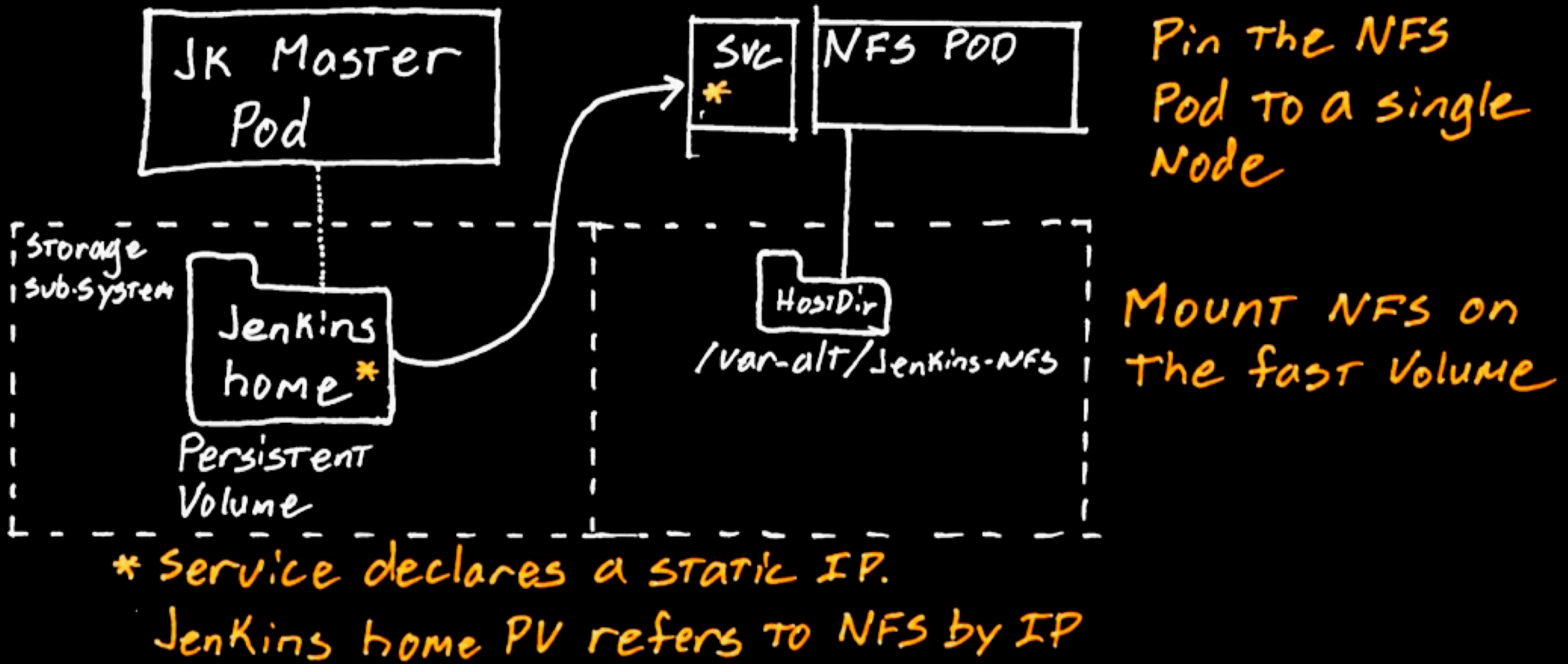
Persistent Storage

on a self-hosted cluster



Persistent Storage

on a self-hosted cluster



Persistent Storage

NFS hosted inside K8s

nfs-server-rc.yaml

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: nfs-server
spec:
  replicas: 1
  selector: ⋯
  template:
    metadata: ⋯
    spec:
      nodeSelector:
        kubernetes.io/hostname: "ip-10-0-129-205"
      containers:
      - name: nfs-server ⋯
      volumes:
      - name: nfs-export-fast
        hostPath:
          path: /var-alt/lib/jenkins-nfs
```

nfs-server-service.yaml

```yaml
kind: Service
apiVersion: v1
metadata:
  name: nfs-server
spec:
  # Declare a static IP since nfs
  # volumes can't use hostname.
  clusterIP: 192.168.129.1
```

nfs-jenkins-home-pv.yaml

```yaml
kind: PersistentVolume
apiVersion: v1
metadata:
  name: jenkins-home-volume
  labels:
    type: local
spec:
  storageClassName: jenkins-home-volume
  capacity:
    storage: 30Gi
  accessModes:
    - ReadWriteMany
  nfs:
    # Use the IP address, hostname doesn't work
    server: 192.168.129.1
    path: "/exports/jenkins-home"
```

Jenkins broke our brains

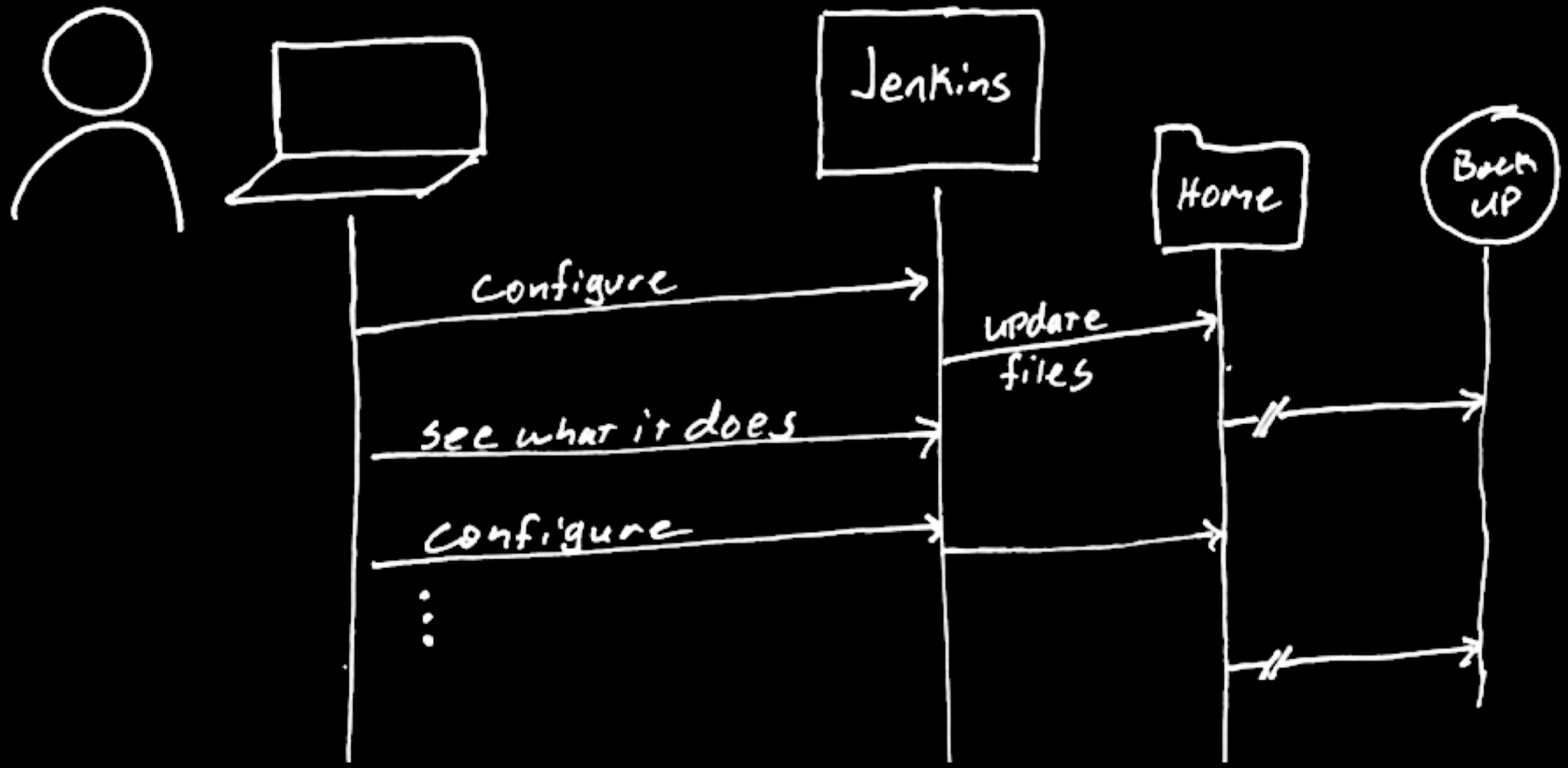
Configure Jenkins to be Efficient and Repeatable

Benchmarking your build VM's IO

Efficient and repeatable
Jenkins configuration

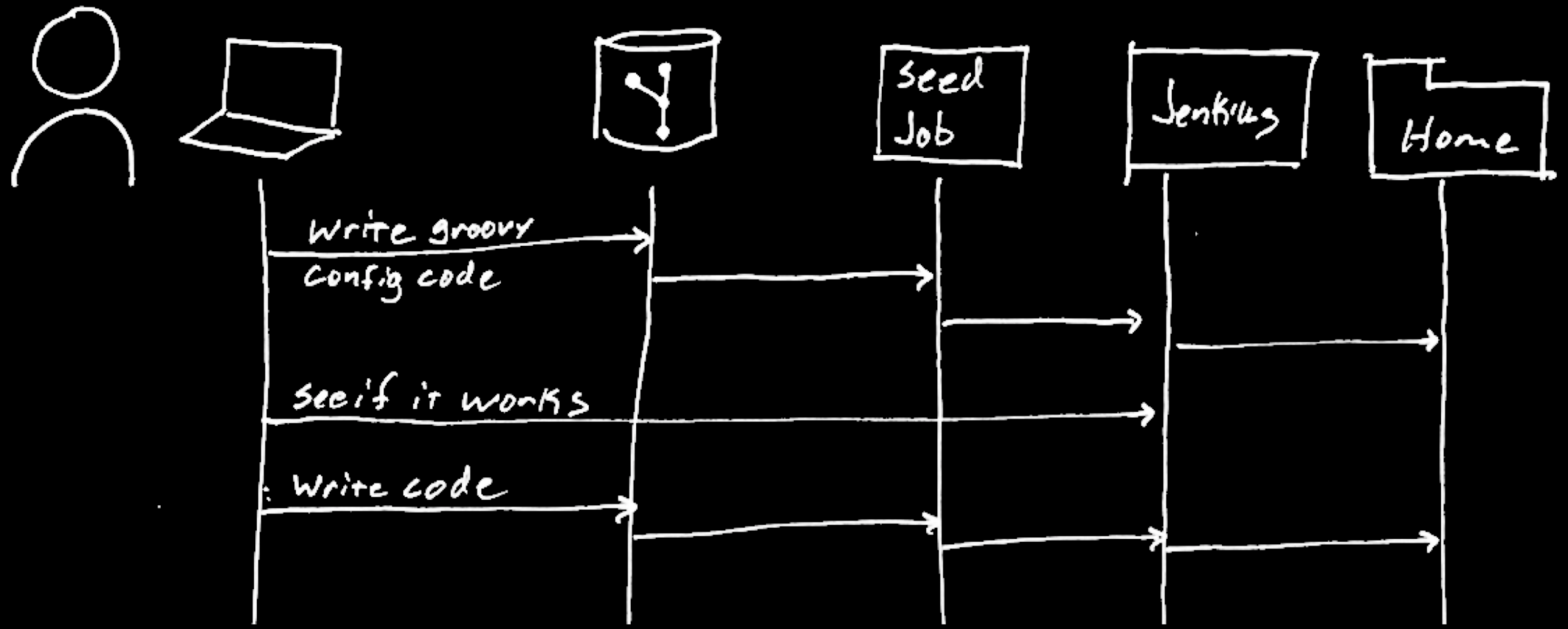
Traditional Style Jenkins Configuration

Hand-configured Jenkins with the Jenkins user interface



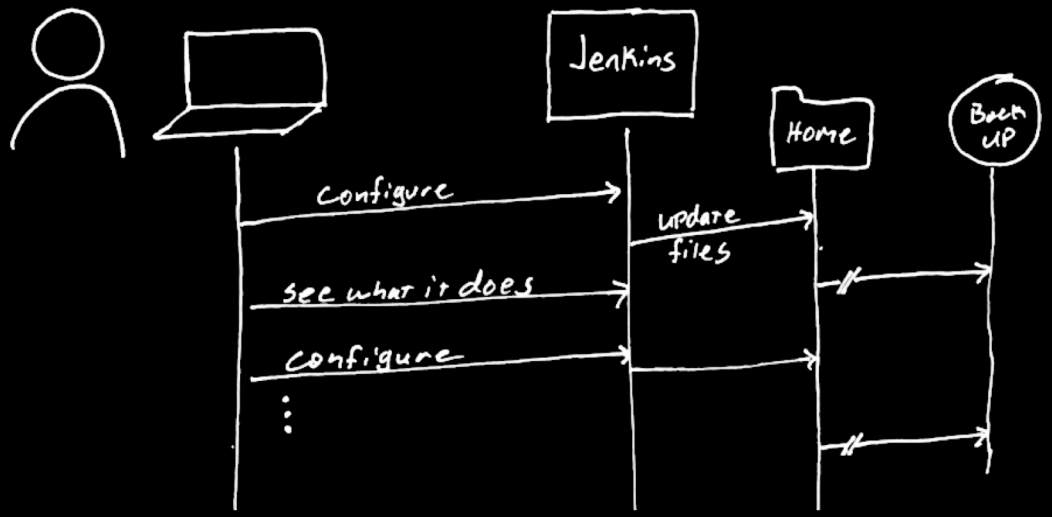
GitOps Style Jenkins Configuration

Jenkins configuration in Jenkinsfile or Groovy DSL

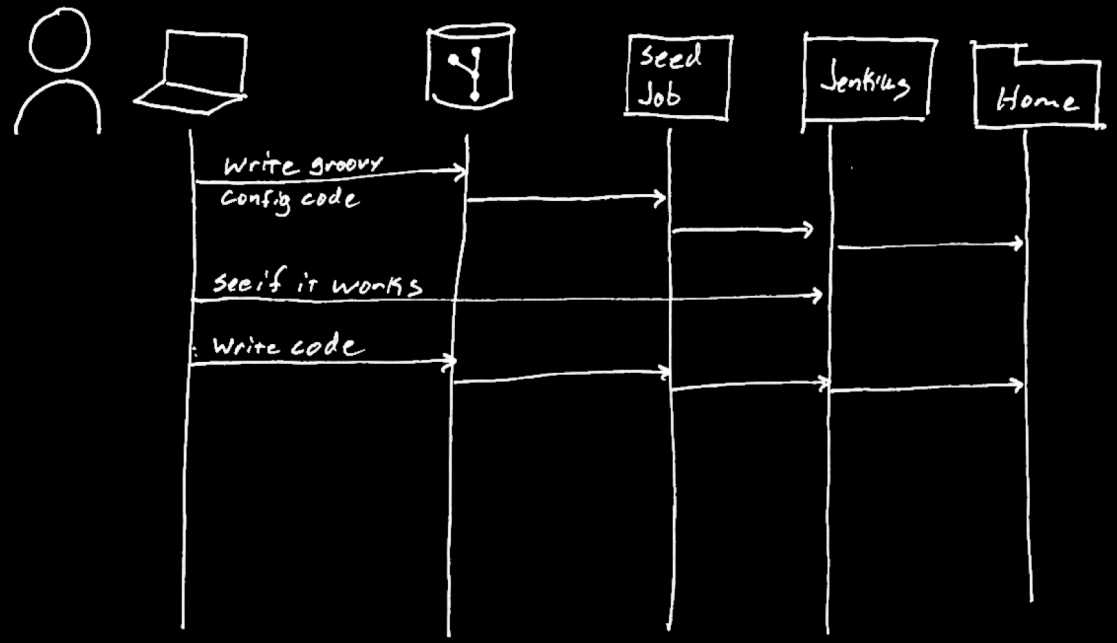


Don’t choose, use both!

Traditional Style for Global Config

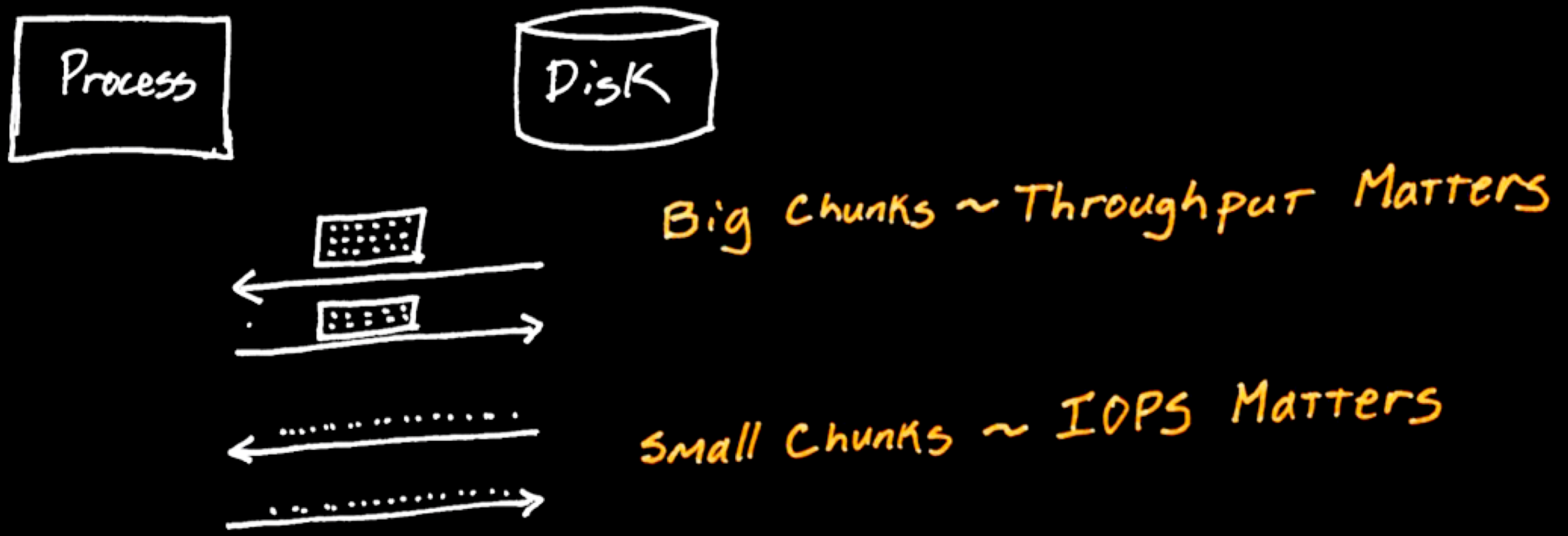


GitOps Style for Project Builds



Performance Benchmarking

A little bit of math can save you a lot of time



```
# Write Big Chunks
time dd if=/dev/zero of=$WORKSPACE/testfile bs=1G count=1 oflag=direct

# Write Small Chunks
time dd if=/dev/zero of=$WORKSPACE/testfile bs=512 count=1000 oflag=direct

# Create Files
STARTMS=$(date +%s%3N)
mkdir $WORKSPACE/sampledir
COUNTER=0
while [  $COUNTER -lt 100 ]; do
  mkdir -p $WORKSPACE/sampledir/$COUNTER
  COUNTER2=0
  while [  $COUNTER2 -lt 100 ]; do
    touch $WORKSPACE/sampledir/$COUNTER/$COUNTER2
    touch $WORKSPACE/sampledir/$COUNTER/$COUNTER2
    let COUNTER2=COUNTER2+1
  done
  let COUNTER=COUNTER+1
done
```


“And then we let developers take it for a drive.”

“Barely working”

Operating Jenkins broke our brains

Builds fail intermittently

OOM

VMs die suddenly

Dangling Docker Berries

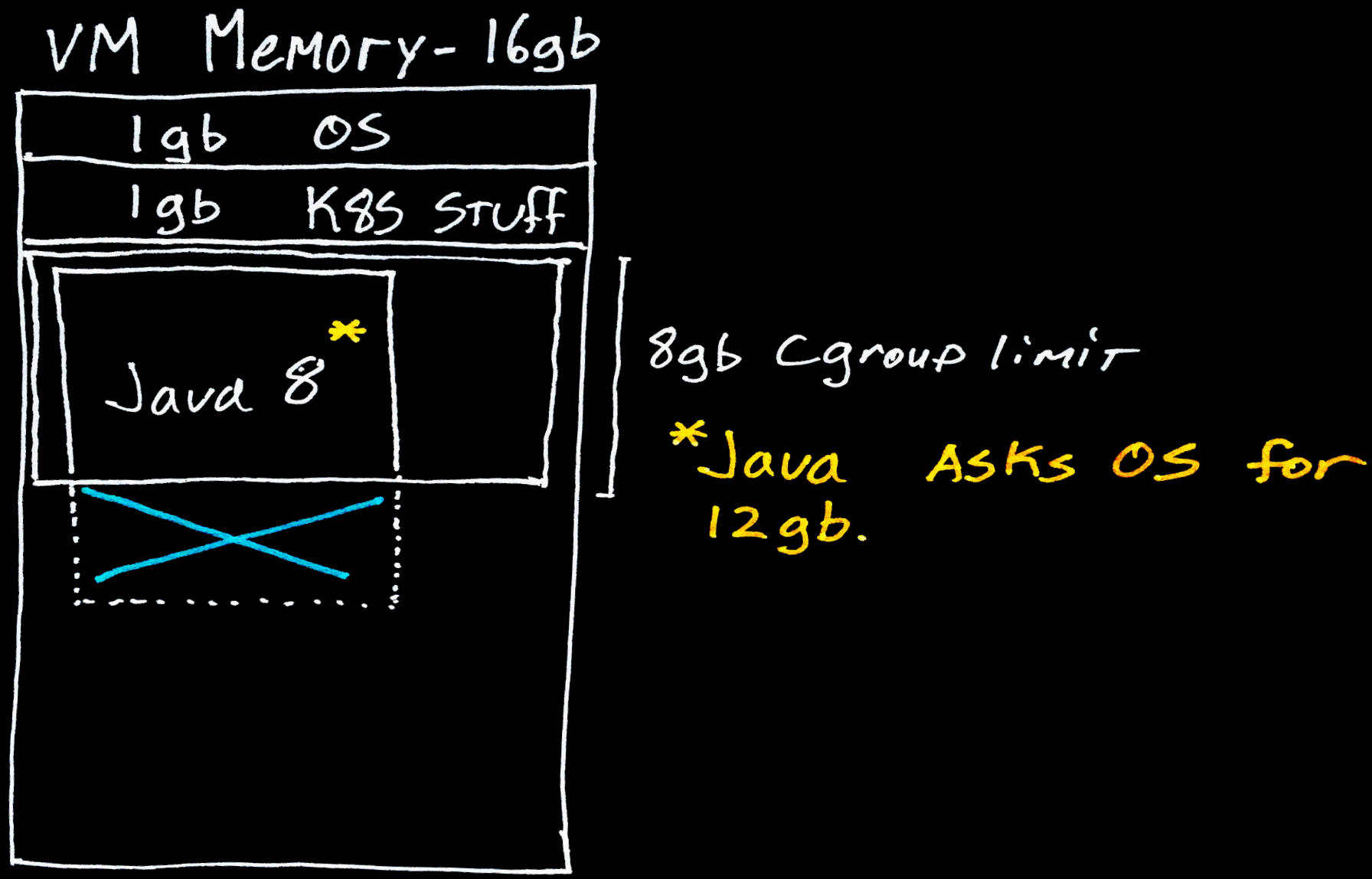
Builds back up endlessly

Pod Tetris

Builds fail intermittently

“What in heck is OOM?”

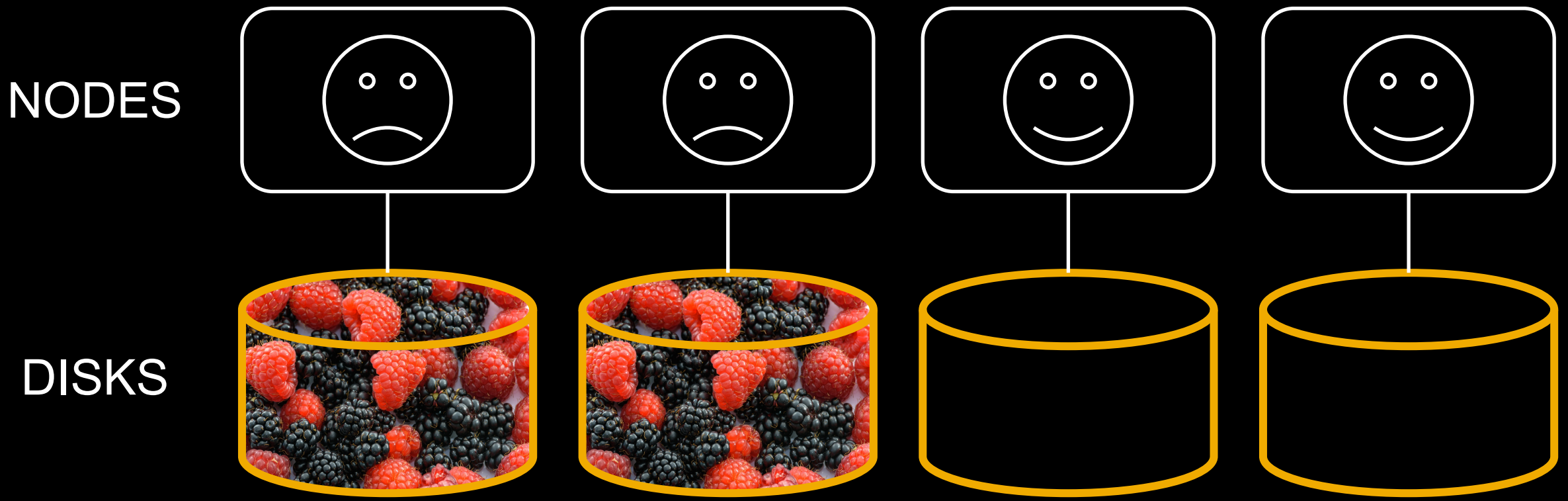
Java doesn't play nice with containers



VMs die suddenly

“What in heck are Dangling DockerBerries?”

Dangling DockerBerries



Prune the DockerBerries hourly

```
# To remove all dangling images
$ docker image prune
# OR
$ docker rmi $(docker images -q -f dangling=true)

# To remove all exited containers
$ docker rm $(docker ps -qa --filter "status=exited")

# To remove dangling volumes
$ docker volume prune
# OR
$ docker volume rm $(docker volume ls  -q --filter dangling=true)
```

Builds back up endlessly

“What in heck is Pod Tetris?”

Pod Tetris

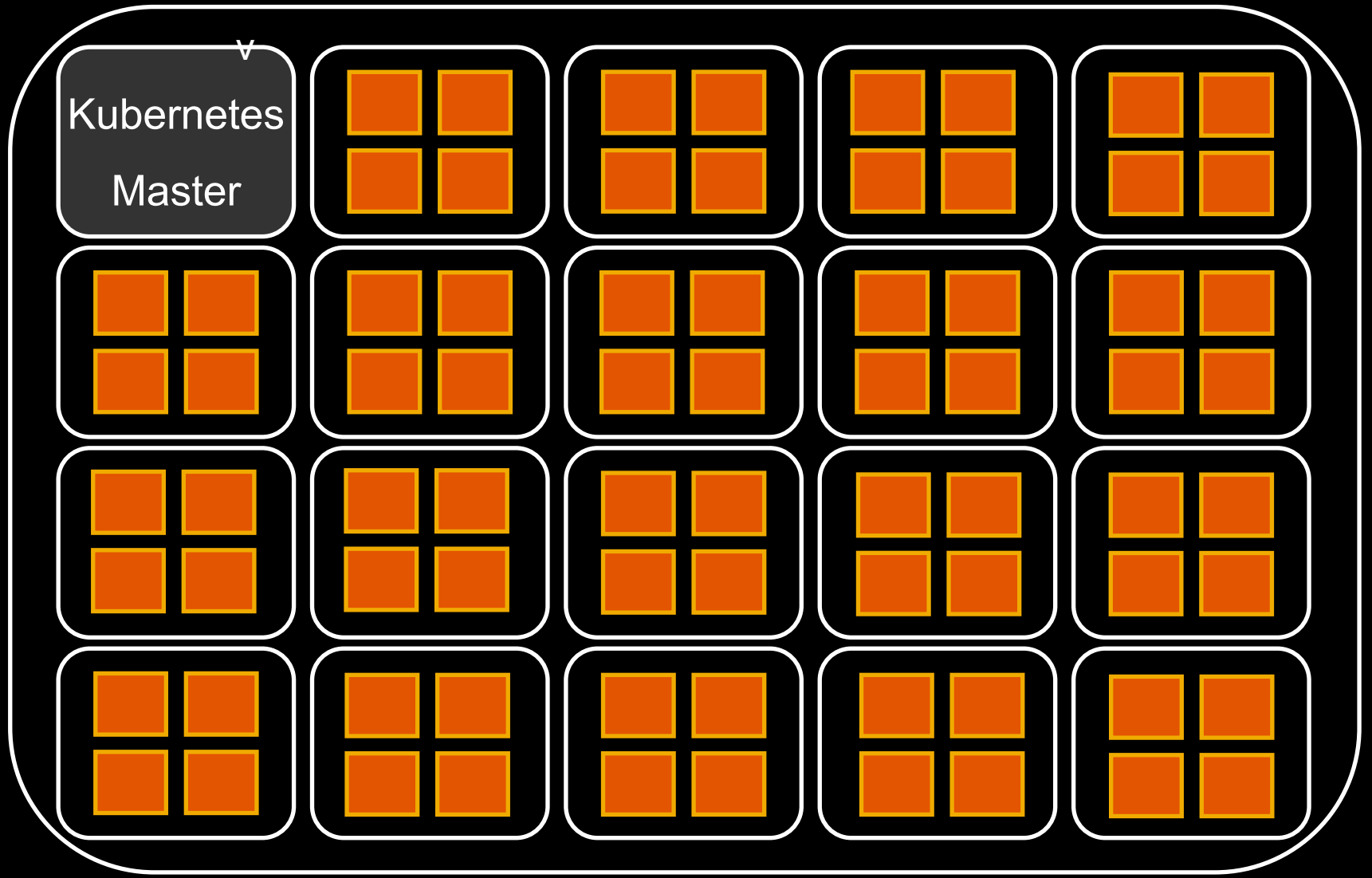
How to lose

Pod count: 23

Cluster used: 100%

(Large pod)
3.5 CPU
12 GB Limit Memory

The entire 24 node K8s cluster



Pod Tetris

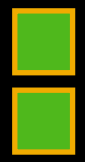
How to WIN

Pod count: 23

Cluster used: 30%

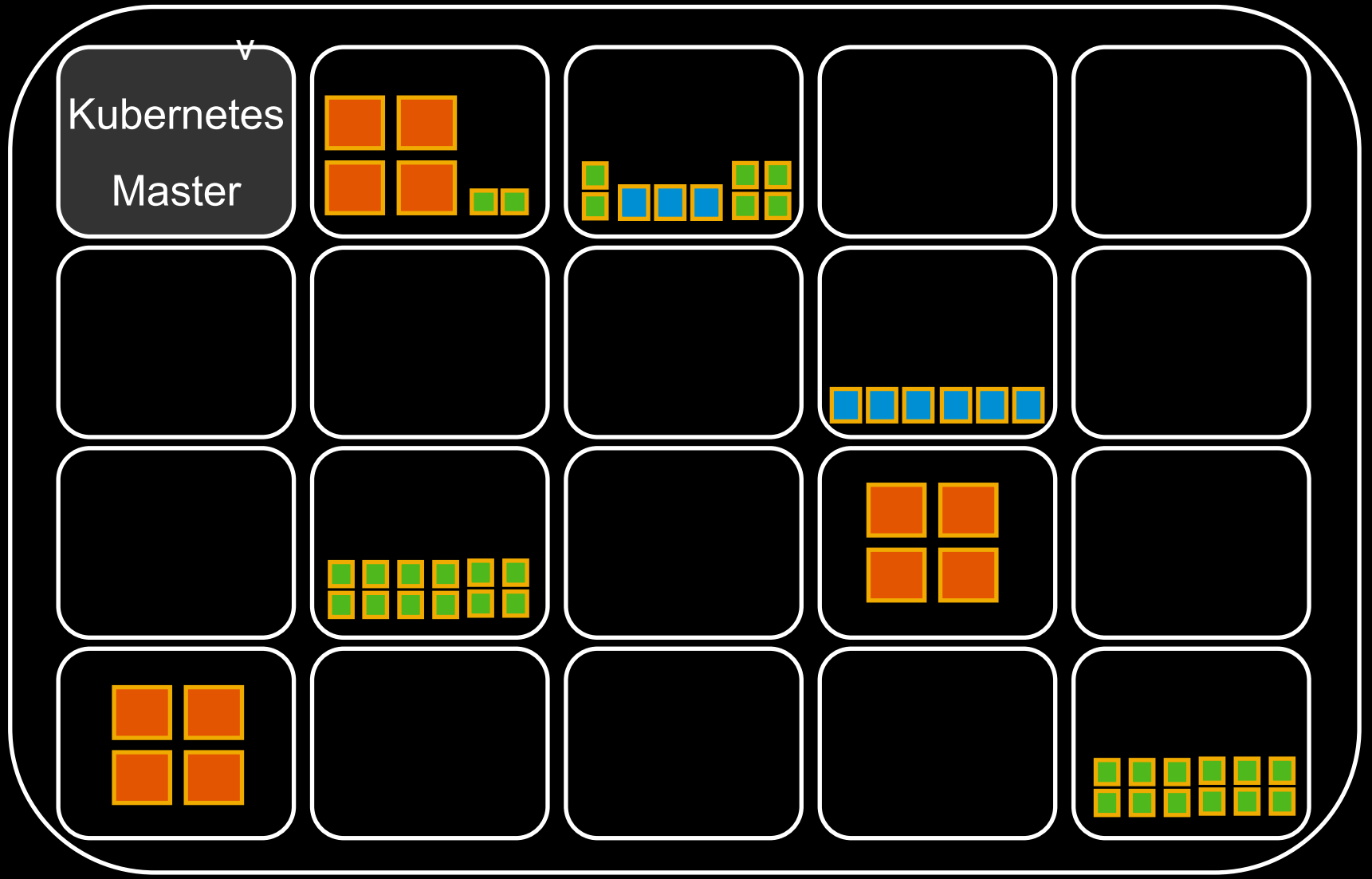
(Large pod)
3 CPU
10 GB Memory

(Small pod)
2 CPU
6 GB Memory



(Tiny pod)
1 CPU
2 GB Memory

The entire 24 node K8s cluster



10-4 good buddy

“Pods are NOT VMs.”

Source: Trout

If you want to do this too

Benchmark IO performance

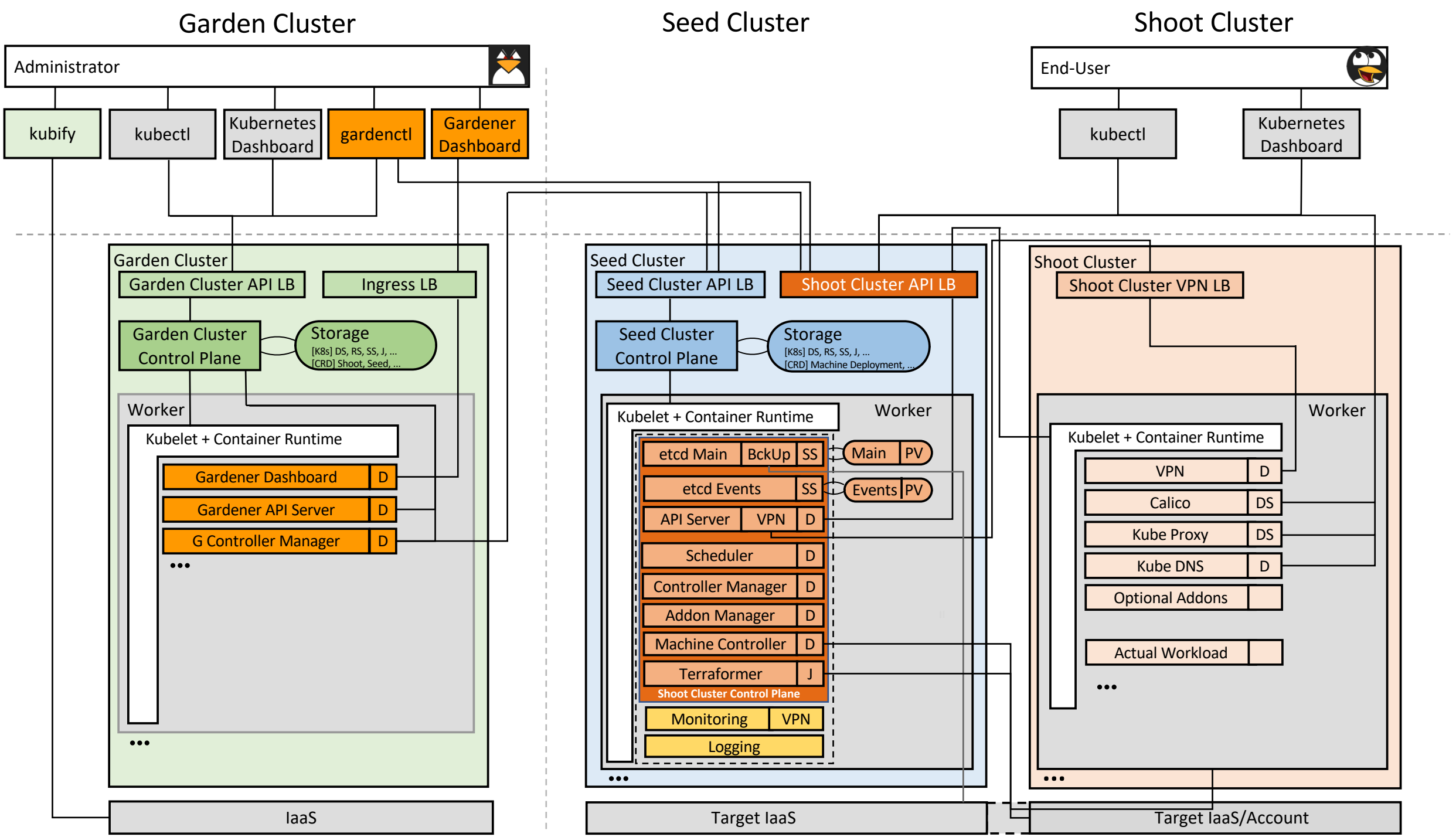
Jenkinsfiles for builds (don’t bother scripting global config)

Play Pod Tetris to Size your jenkins

Don’t try to build Kubernetes yourself (consider Gardener)

Gardener

Open source Kubernetes management from SAP


Garden Cluster
Seed Cluster
Shoot Cluster
Administrator
kubify
kubectl
Kubernetes Dashboard
gardenctl
Gardener Dashboard
End-User
kubectl
Kubernetes Dashboard
Garden Cluster
Garden Cluster API LB
Ingress LB
Garden Cluster Control Plane
Storage
[K8s] DS, RS, SS, J, ...
[CRD] Shoot, Seed, ...
Worker
Kubelet + Container Runtime
Gardener Dashboard D
Gardener API Server D
G Controller Manager D
IaaS
Seed Cluster
Seed Cluster API LB
Shoot Cluster API LB
Seed Cluster Control Plane
Storage
[K8s] DS, RS, SS, J, ...
[CRD] Machine Deployment, ...
Kubelet + Container Runtime
Worker
etcd Main BckUp SS
Main PV
etcd Events SS
Events PV
API Server VPN D
Scheduler D
Controller Manager D
Addon Manager D
Machine Controller D
Terraformer J
Shoot Cluster Control Plane
Monitoring VPN
Logging
Target IaaS
Shoot Cluster
Shoot Cluster VPN LB
Worker
Kubelet + Container Runtime
VPN D
Calico DS
Kube Proxy DS
Kube DNS D
Optional Addons
Actual Workload
Target IaaS/Account

SAP has a booth

Learn more about our projects on Kubernetes from SAP

Gardener

Manage Kubernetes clusters as a service at scale

Monitor repair, patch or upgrade your clusters in an automated way

Central dashboard that simplifies administrative tasks across cloud infrastructures
gardener.cloud/

Kyma

Connect and extend products using serverless computing and microservices architecture

Customize your solution using the technology you want

Build an end-to-end customer experience using your existing technology
kyma-project.io/

Thank you SAP

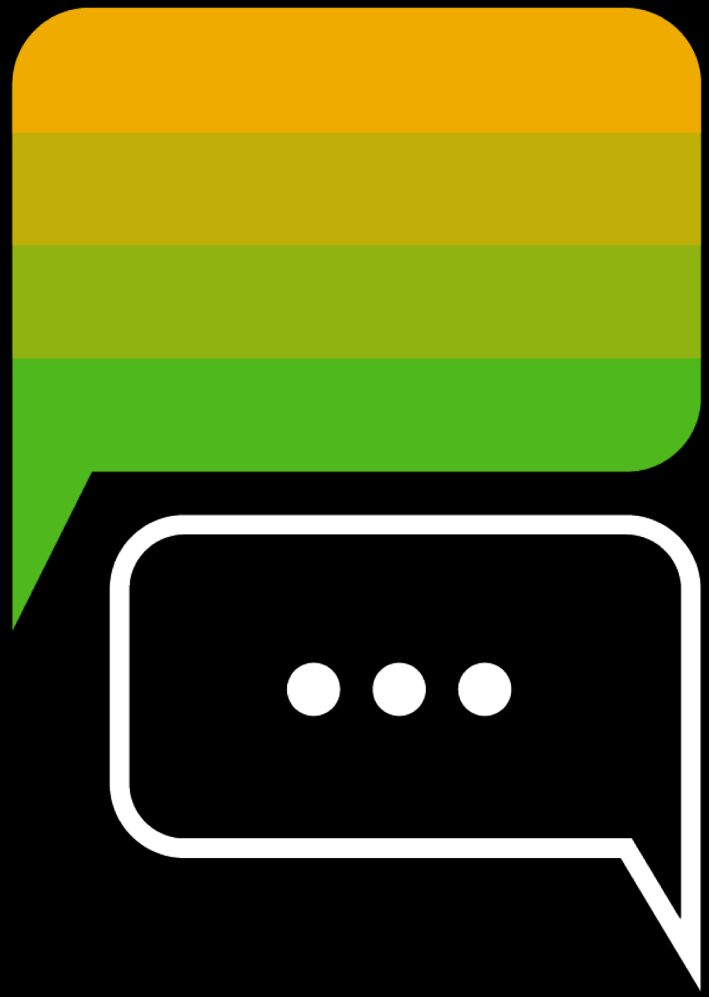
Open source foundations and noteworthy projects by SAP

Gardener

Kyma

Service Manager

Questions & Answers



Loren Trout
Devops Engineer, SAP
Loren.trout@sap.com
@enzotrout

Jonathan Hess
Software Engineer, SAP
jonathan.hess@sap.com
@solarhess

github.com/solarhess/jenkins_kube_brains

Images attributions

"Fury Truck" by pxhere is licensed under CC BY 2.0

"Jenkins" by Jenkis.io is licensed under CC BY 4.0

"Runaway Truck Ramp - Monarch Pass" by Larry Lamsa is licensed under CC BY 2.0

"JunkYard" by Tom Fisk is licensed under pexels licence

"Truck into Mountains" by 500photos is licensed under pexels licence

"Berries" by **PhotoMIX** is licensed under pexels licence