



COMMUNITY DAY

Kubernetes at AWS

Past Present and Future

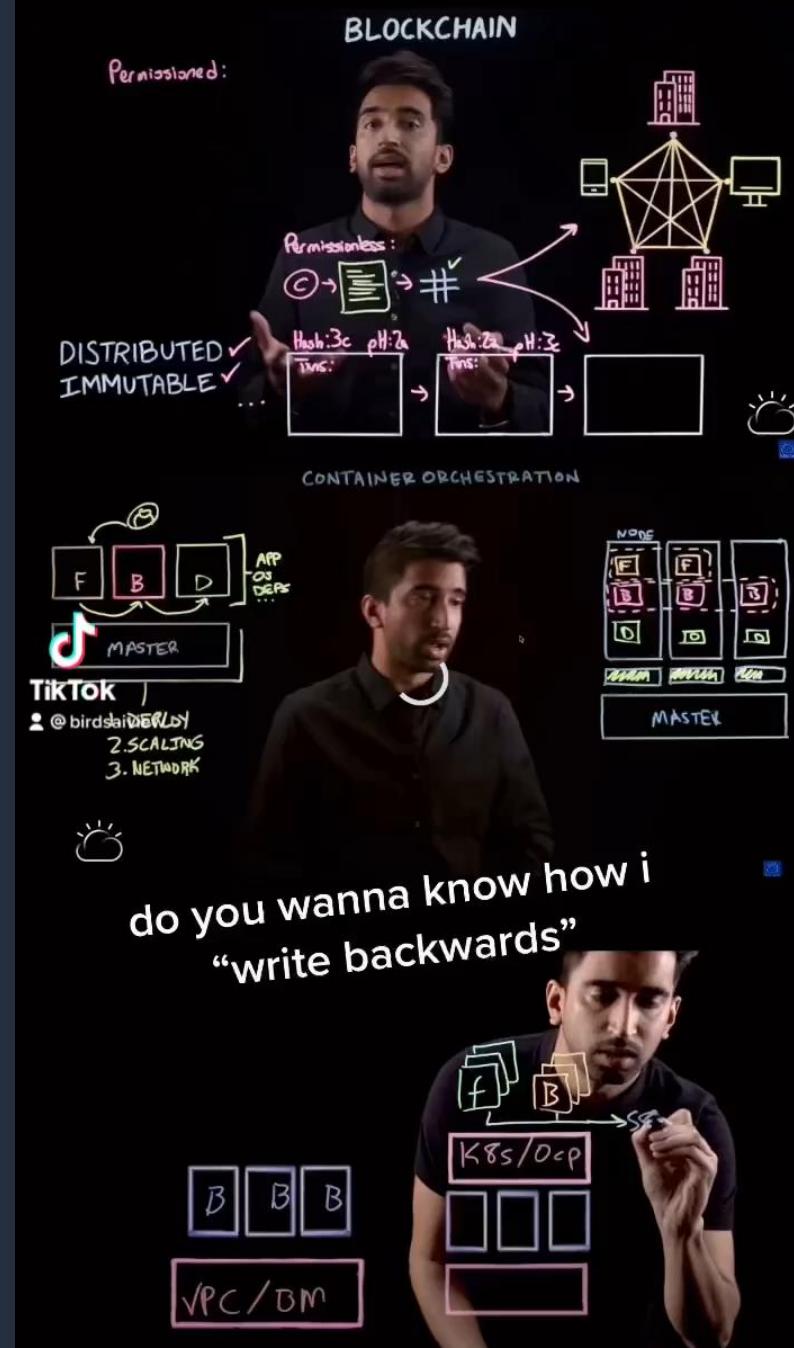
Sai Vennam

Principal Developer Advocate - AWS

aws
COMMUNITY DAY
PUNE

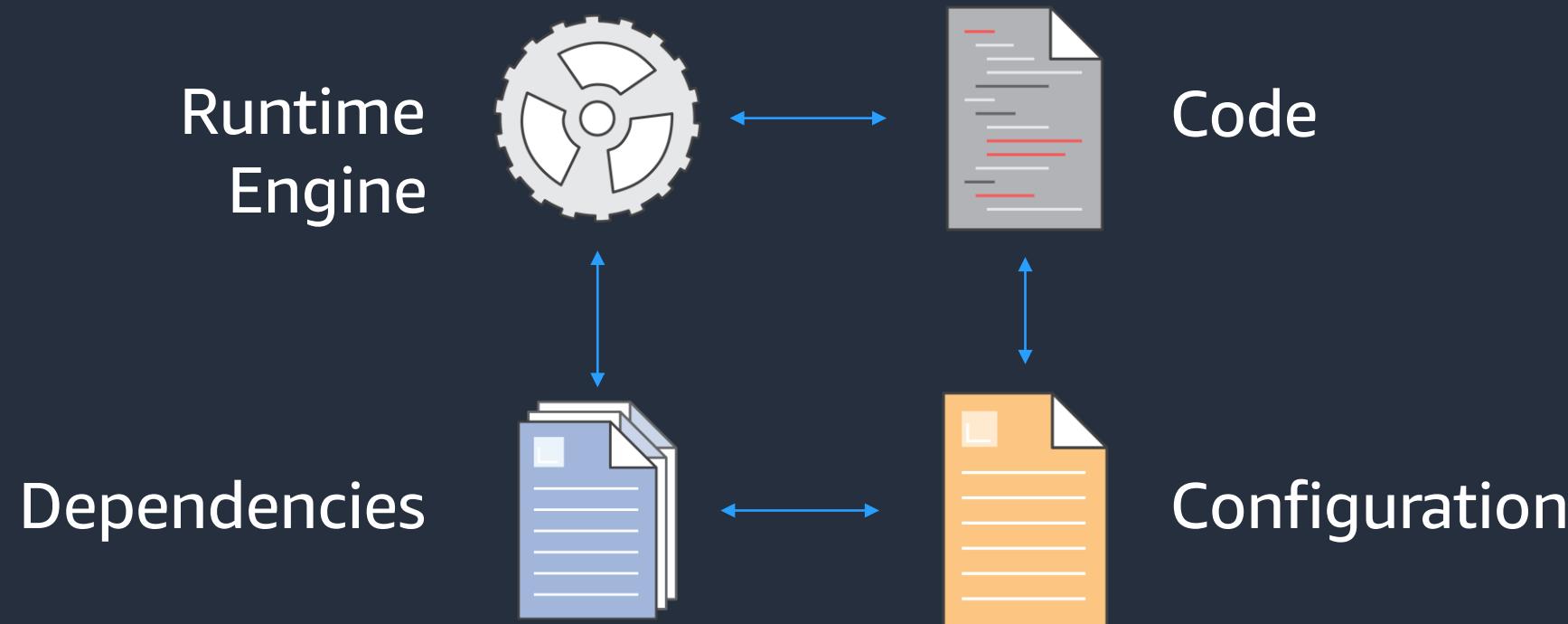
A little about me...

- Born in India
- Ex-IBM
- I record and produce all my own videos!
 - Check out **Containers from the Couch** on YT (cftc.info)
- I love cars, smart-home gadgets, PC building, sim-racing and live music

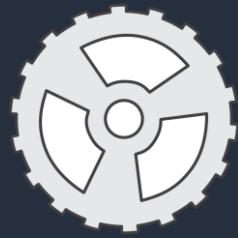


Why are companies moving to containers?

Application environment components



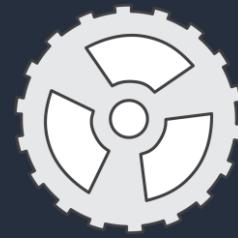
It worked on my machine, why not in prod?



v6.0.0



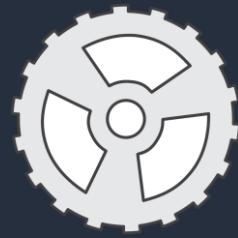
Local Laptop



v7.0.0



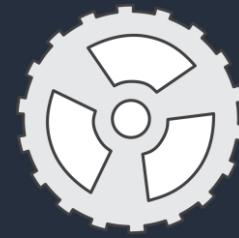
Staging / QA



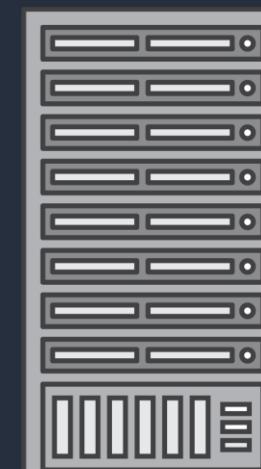
v4.0.0



Production



v7.0.0



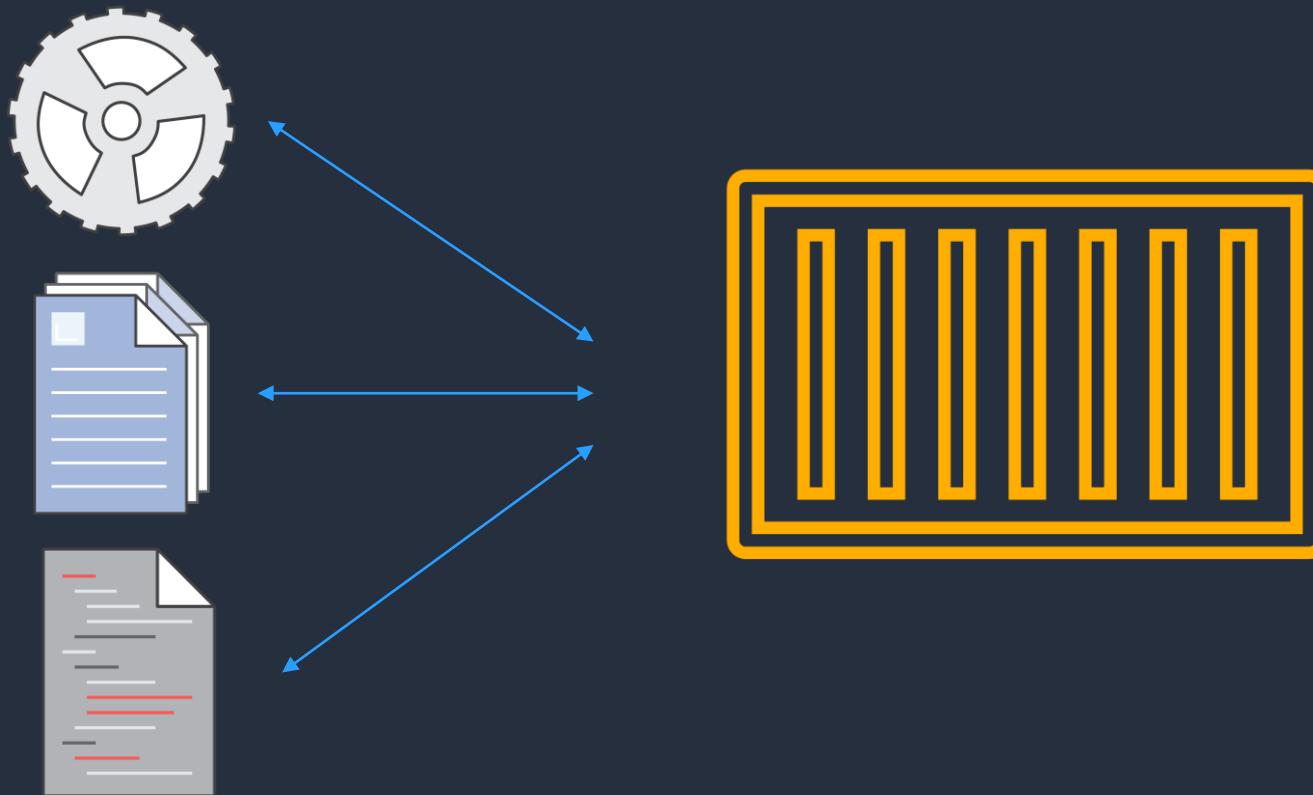
On-Prem

Containers to the rescue

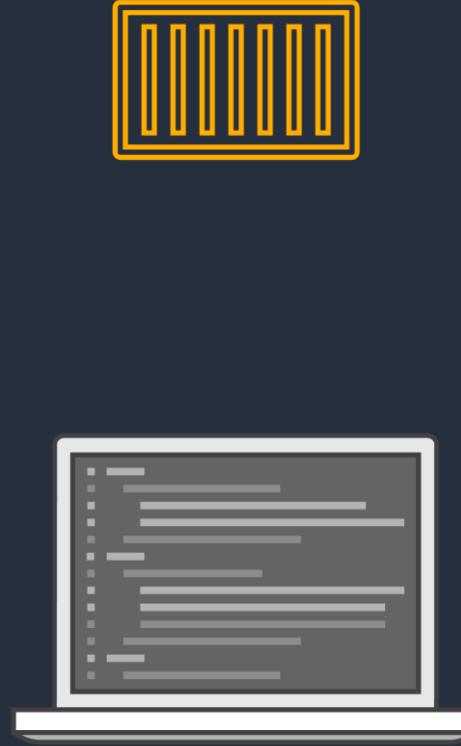
Runtime
Engine

Dependencies

Code



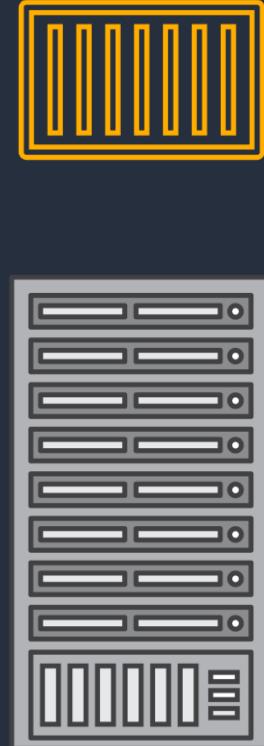
Four environments, same container



Local Laptop



Staging / QA



Production



On-Prem

Why customers adopt containers



Standardize IT operations in order to accelerate delivery and changes
rapid change == innovation



Reduce fixed expense
eliminate complex contracts and management overhead



Enable the entire organization
support multiple environments and different use cases



Plan for the future and reduce risk
standards enable hiring and long-term development efforts

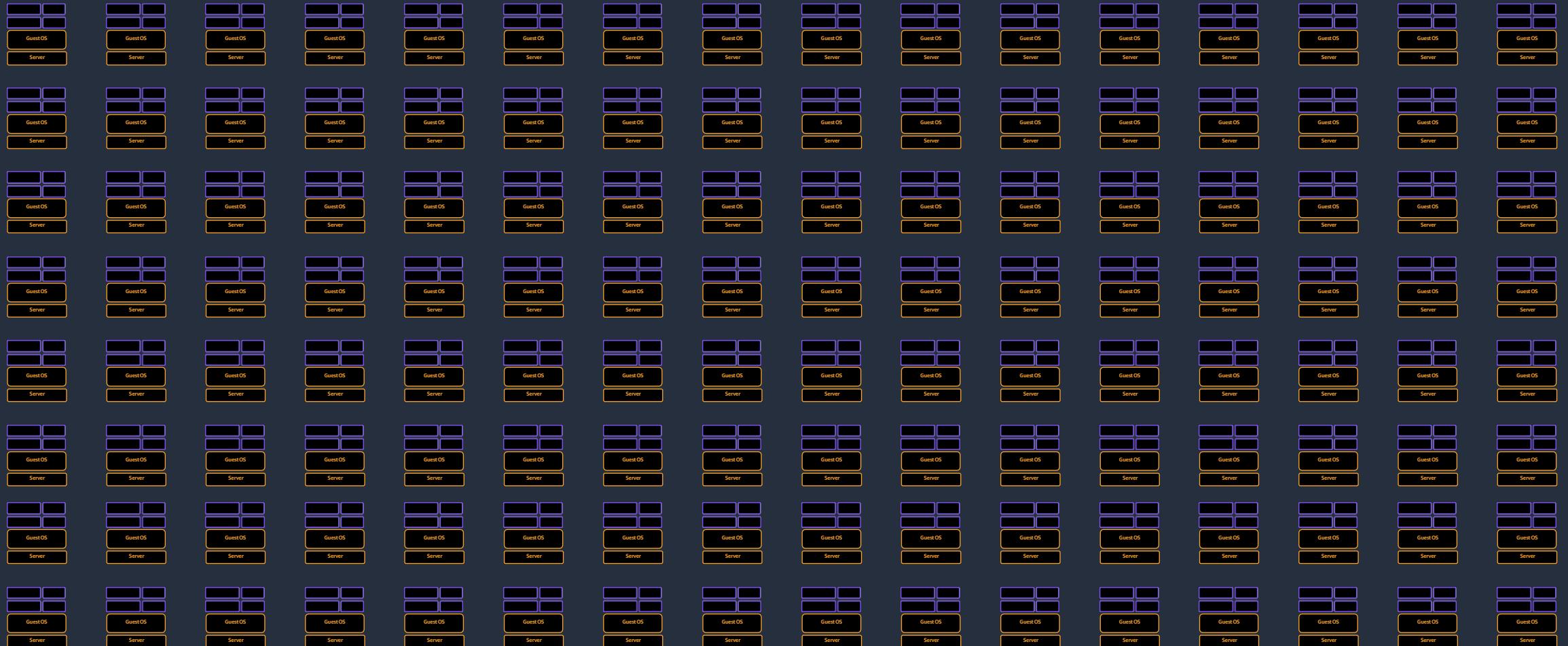
A few hosts?



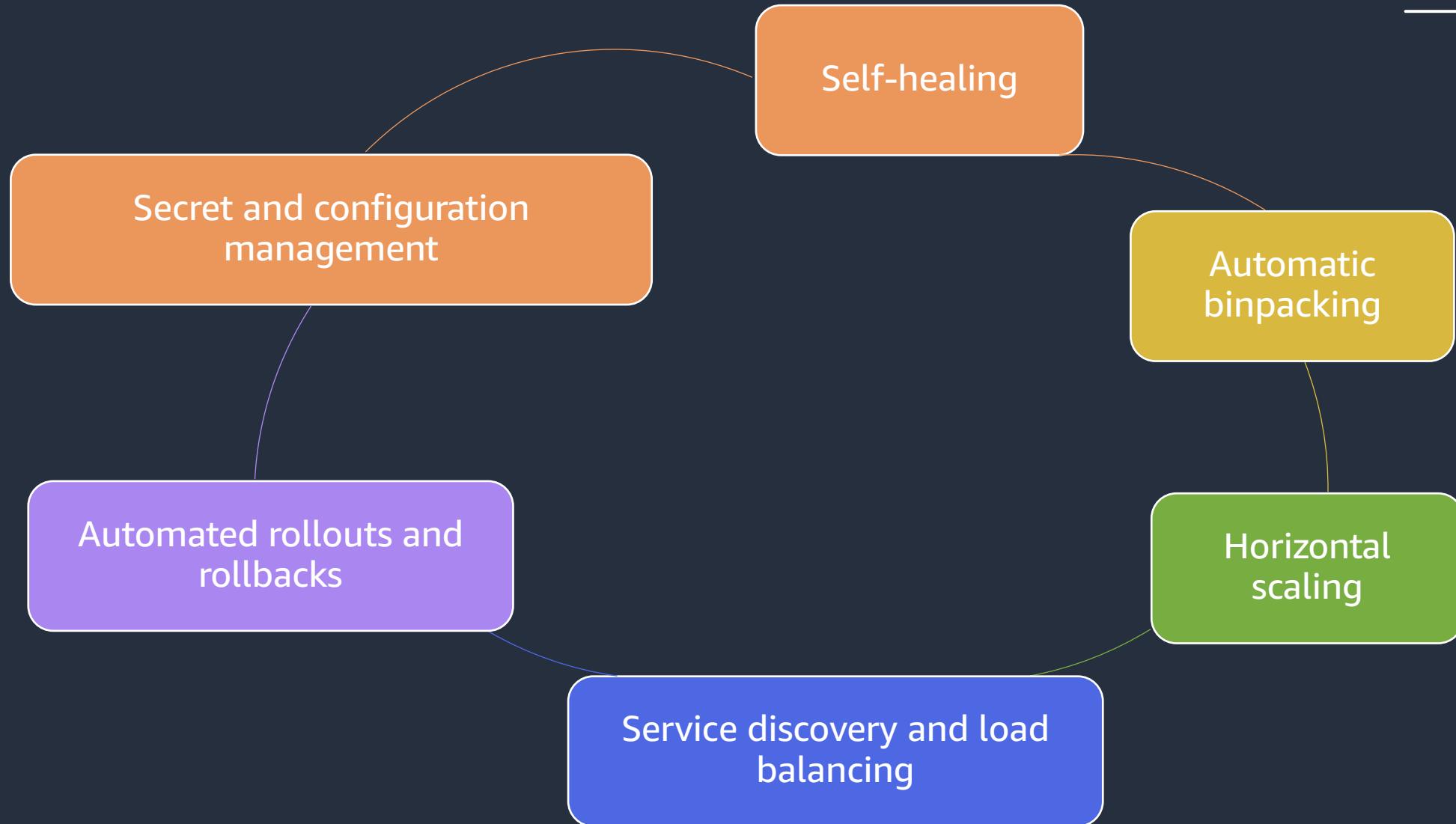
More Hosts?



Lots of hosts!



Orchestrating containers



“More than 2/3 of
containers in the
Cloud run on AWS”

CNCF survey

Kubernetes

aws
COMMUNITY DAY
— PUNE —



Kubernetes



Amazon EKS

Amazon EKS is the most trusted and secure way to run Kubernetes



Amazon EKS



EKS runs vanilla Kubernetes. EKS is upstream and certified conformant version of Kubernetes (with backported security fixes).



EKS supports last 4 versions of Kubernetes, giving customers time to test and roll out upgrades.



EKS provides a managed Kubernetes experience for performant, reliable, and secure Kubernetes.



EKS makes Kubernetes operations, administration, and management simple and boring.

AWS is the **best place to run Kubernetes. 65% of organizations choose AWS to run their containers.**

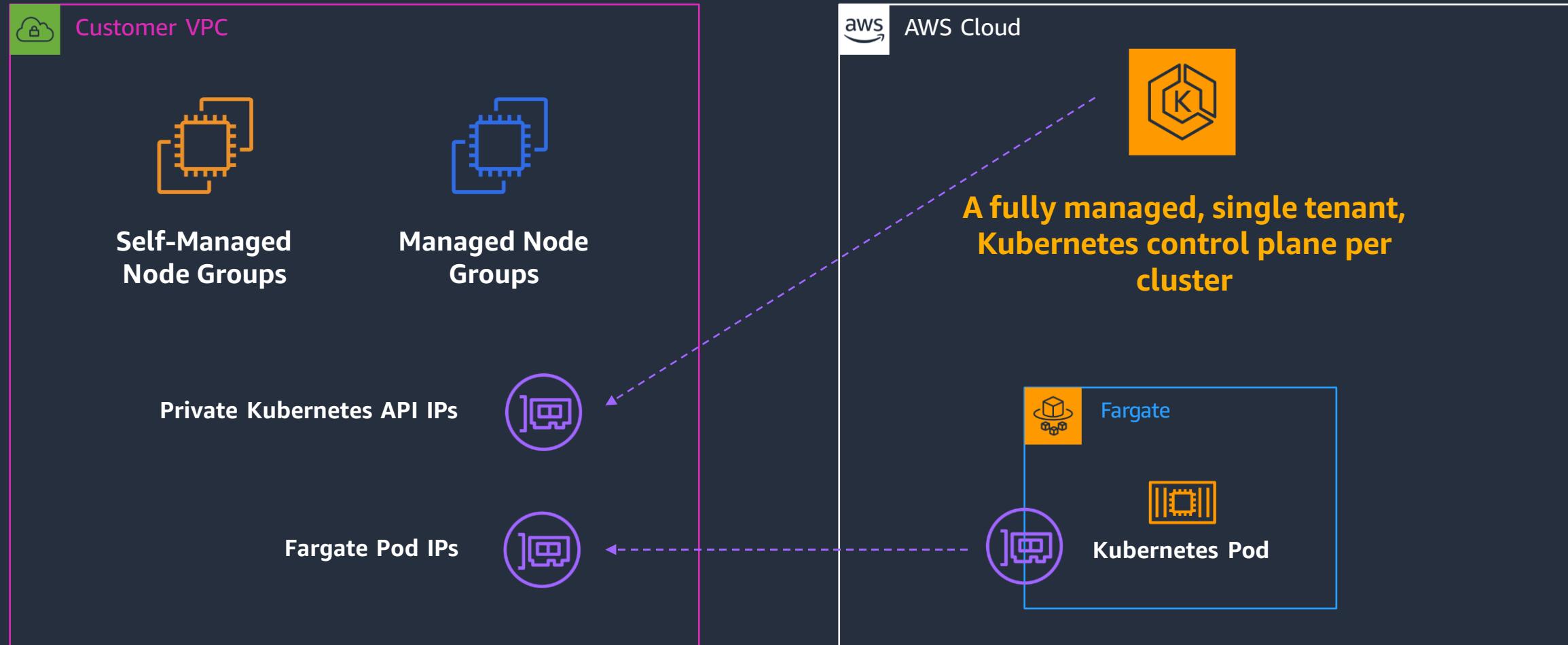


[CNCF State of Cloud Native Development](#)



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS High Level Architecture



EKS enables you to focus on strategic operations rather than undifferentiated heavy lifting

Strategic Operations



Product Releases



Application Security



Organizational Structures



Software Supply Chain



Customer Connectivity

Tactical Operations



Control Planes



Scaling + Availability



Infrastructure Reliability



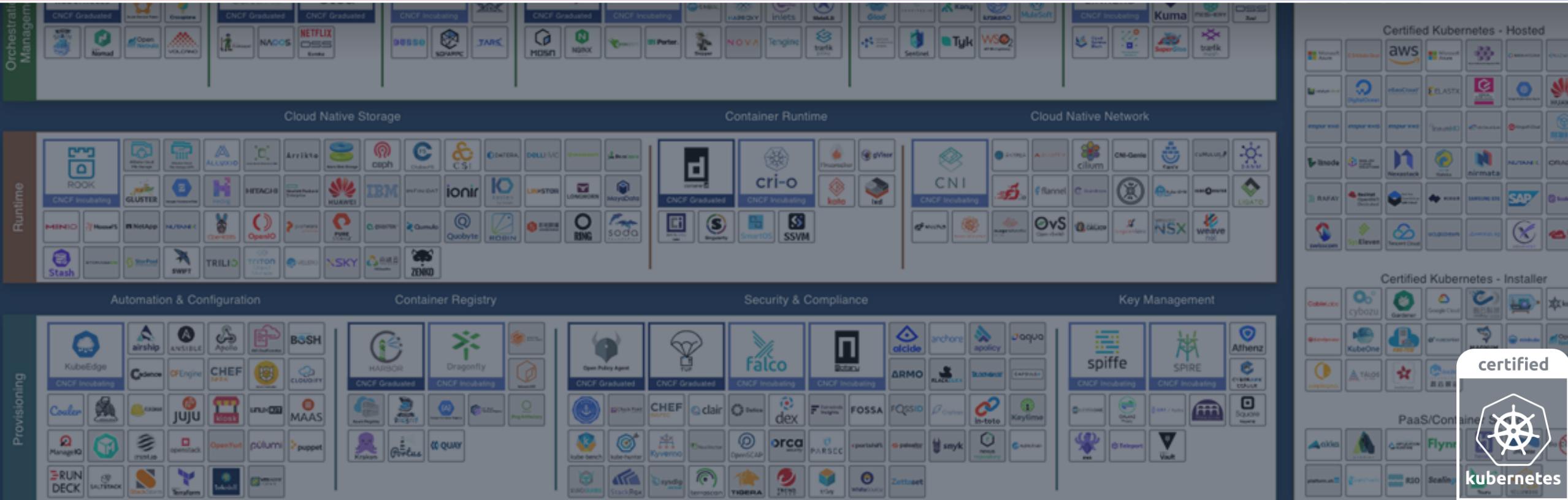
Authentication + Security



Backups and Resiliency

Native and upstream

**If it runs on Kubernetes, it runs on Amazon EKS.
That includes Istio, Knative, Kubeflow, etc.**



AWS is committed to supporting the Kubernetes community regardless of how customers choose to run Kubernetes

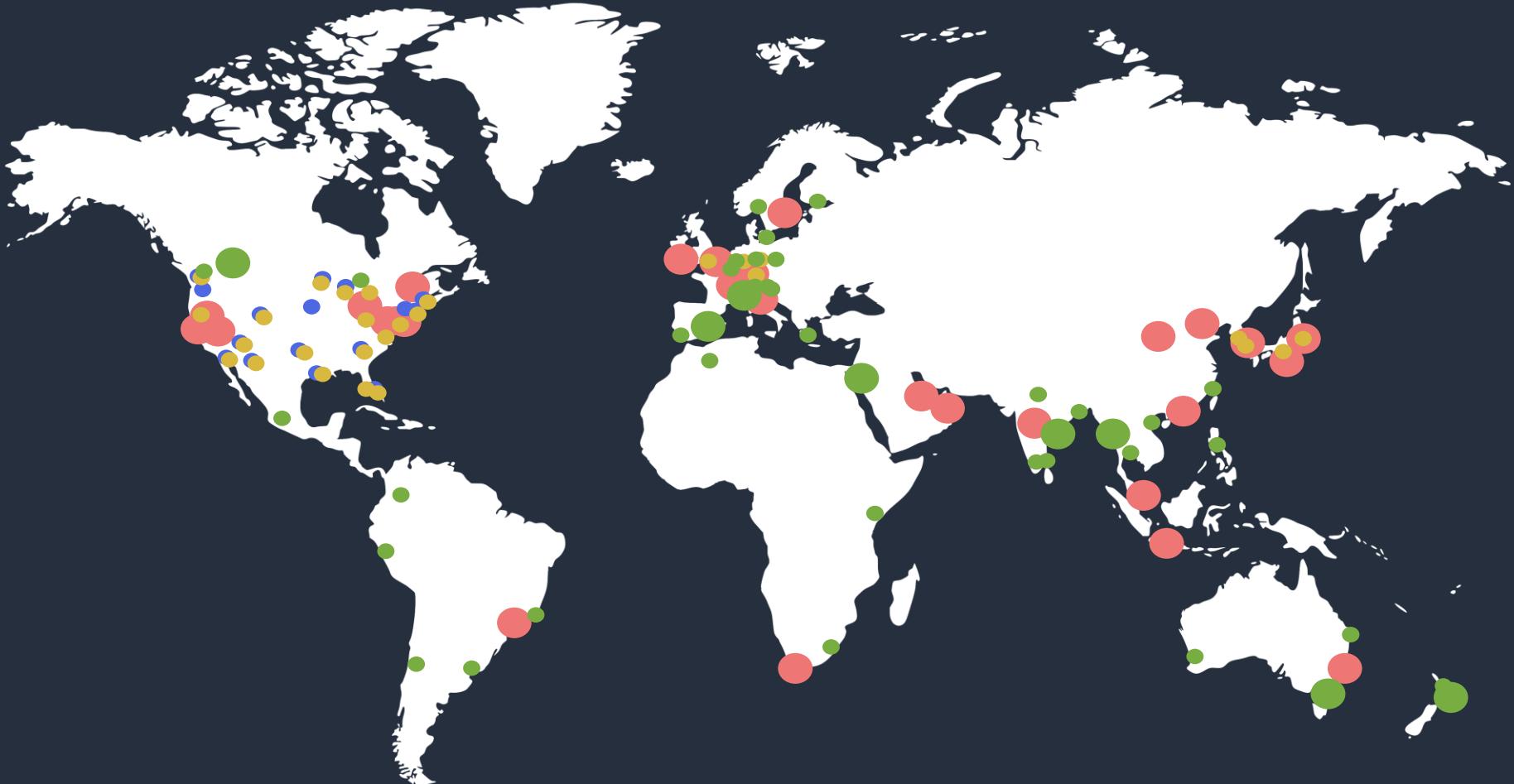


Security is job zero at AWS and that includes Kubernetes. **AWS is a member of the Product Security Committee**, who are responsible for the security the Kubernetes project.

-  kubernetes/cloud-provider-aws
-  kubernetes/autoscaler
-  kubernetes-sigs/aws-ebs-csi-driver
-  kubernetes-sigs/aws-efs-csi-driver
-  kubernetes-sigs/aws-fsx-csi-driver
-  kubernetes-sigs/aws-iam-authenticator
-  kubernetes-sigs/aws-load-balancer-controller
-  kubernetes-sigs/aws-encryption-provider
-  awslabs/karpenter
-  aws/eks-distro

CNI, CSI, Kubernetes, and many more...

Amazon global reach



30 geographic regions

96 availability zones

21 local zones

29 wavelength zones

Coming soon

5 new geographic regions

15 new availability zones

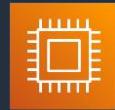
30 new local zones

Portfolio Deployment Options



COMMUNITY DAY

PUNE



Amazon EKS in AWS

Best, scale, performance, & cost

AWS managed



Amazon EKS +
AWS Local Zones
AWS Wavelengths

Fully managed, latency optimized



Amazon EKS + AWS Outposts

Fully managed experience where you need it



Amazon EKS + AWS SNOW

AWS hardware and support at the edge



Amazon EKS Anywhere

Customer hardware, AWS supported tooling & software



Amazon EKS Distro

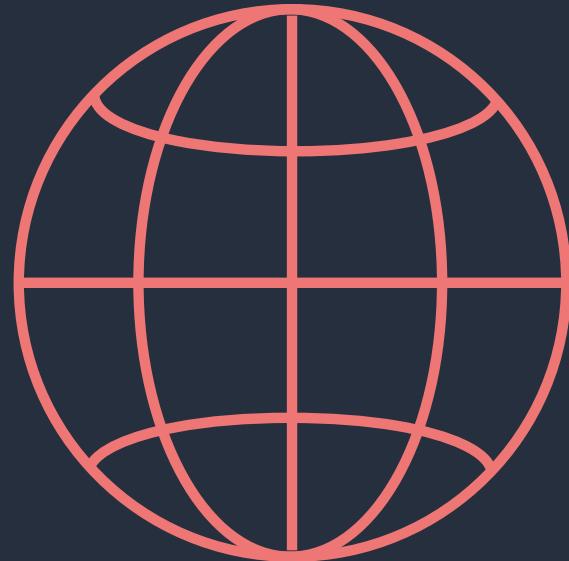
Customer hardware, customer management tools



Customer managed



Run everywhere



Ohio, Virginia, California, Oregon, Cape Town,
Hong Kong, **Hyderabad**, Mumbai, Osaka,
Seoul, Singapore, Sydney, Tokyo, Jakarta,
Canada, Beijing, Ningxia, Frankfurt, Ireland,
London, Paris, **Spain**, Milan, Paris, Switzerland,
Stockholm, Bahrain, São Paulo, GovCloud East
& West

Local Zones, Wavelengths, Outposts, **Snow**,
Your Data Center

Key area of focus

COST OPTIMIZATION

Customer cost challenges



Allocate costs across teams and departments

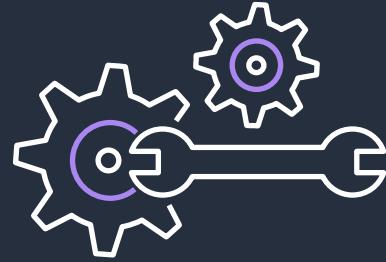


Chargeback/showback



Reporting, budget forecasting, cost optimization

Karpenter – cost efficient compute for Kubernetes



Karpenter is an **intelligent** and **high-performance** Kubernetes compute provisioning and management solution



Karpenter lets you take **full advantage of AWS** with its deep integration between Kubernetes and Amazon EC2

Why Karpenter?



Responds in seconds
when application load
changes **improving
availability**



Kubernetes-native
CRDs and opinionated
defaults **minimize
operational overhead**



Intelligently chooses
instance types and
consolidates pods to
lower compute costs

Karpenter in Action

DEMO



```
./monitui (monitui)          961          bash (vi)
bash-5.1$ kubectl get node
NAME                               STATUS  ROLES   AGE    VERSION
ip-192-168-84-35.us-west-2.compute.internal  Ready   <none>  9d    v1.23.9-eks-ba74326
bash-5.1$ kubectl get pod -A
NAMESPACE      NAME                READY   STATUS    RESTARTS   AGE
karpenter     carpenter-65f6584977-tqbr5  2/2     Running   0          8m27s
karpenter     carpenter-65f6584977-xjzk7  2/2     Running   0          8m27s
kube-system   aws-load-balancer-controller-5dc48f896d-vrf6b  1/1     Running   1 (8d ago)  8d
kube-system   aws-load-balancer-controller-5dc48f896d-vrv2p  1/1     Running   0          8d
kube-system   aws-node-76tzc            1/1     Running   0          9d
kube-system   coredns-85d5b4454c-fhxwb   1/1     Running   0          8d
kube-system   coredns-85d5b4454c-m4cpp   1/1     Running   0          8d
kube-system   efs-csi-controller-69bcf9b47-swml4   3/3     Running   1 (8d ago)  8d
kube-system   efs-csi-controller-69bcf9b47-x87xz   3/3     Running   0          8d
kube-system   kube-proxy-qd8dj        1/1     Running   0          9d
kube-system   metrics-server-64cf6869bd-v29ln   1/1     Running   0          8d
monitoring    prometheus-node-exporter-2b7rl   1/1     Running   0          9d
bash-5.1$ vi inflate.yaml
bash-5.1$
```

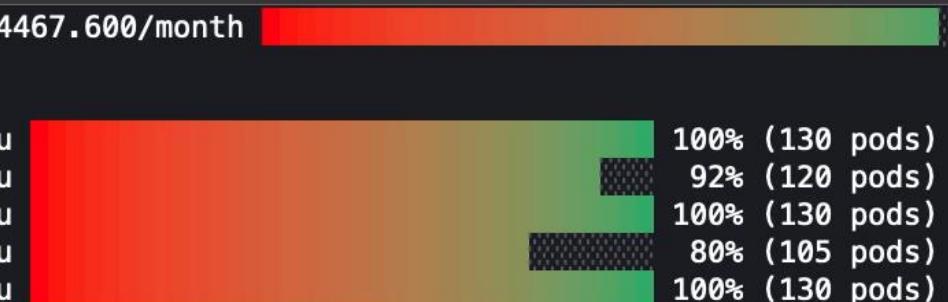
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: inflate
spec:
  replicas: 0
  selector:
    matchLabels:
      app: inflate
  template:
    metadata:
      labels:
        app: inflate
    spec:
      terminationGracePeriodSeconds: 0
      containers:
        - name: inflate
          image: public.ecr.aws/eks-distro/kubernetes/pause:3.2
          resources:
            requests:
              cpu: 250m

~
```

```
./monitui (monitui)          861          bash (vi)
bash-5.1$ kubectl get node
NAME                               STATUS  ROLES   AGE    VERSION
ip-192-168-84-35.us-west-2.compute.internal  Ready   <none>  9d    v1.23.9-eks-ba74326
bash-5.1$ kubectl get pod -A
NAMESPACE      NAME                READY  STATUS  RESTARTS  AGE
karpenter     karpenter-65f6584977-tqbr5  2/2    Running  0          8m27s
karpenter     karpenter-65f6584977-xjzk7  2/2    Running  0          8m27s
kube-system   aws-load-balancer-controller-5dc48f896d-vrf6b  1/1    Running  1 (8d ago)  8d
kube-system   aws-load-balancer-controller-5dc48f896d-vrv2p  1/1    Running  0          8d
kube-system   aws-node-76tzc            1/1    Running  0          9d
kube-system   coredns-85d5b4454c-fhxwb  1/1    Running  0          8d
kube-system   coredns-85d5b4454c-m4cpp  1/1    Running  0          8d
kube-system   efs-csi-controller-69bcf9b47-swml4  3/3    Running  1 (8d ago)  8d
kube-system   efs-csi-controller-69bcf9b47-x87xz  3/3    Running  0          8d
kube-system   kube-proxy-qd8dj        1/1    Running  0          9d
kube-system   metrics-server-64cf6869bd-v29ln  1/1    Running  0          8d
monitoring    prometheus-node-exporter-2b7rl  1/1    Running  0          9d
bash-5.1$ vi inflate.yaml
bash-5.1$
```

bash (bash) bash (bash)

./monitui (monitui)
5 nodes 150500m/159250m 94.5% cpu \$6.120/hour \$4467.600/month
615 pods (0 pending 615 running 615 bound)



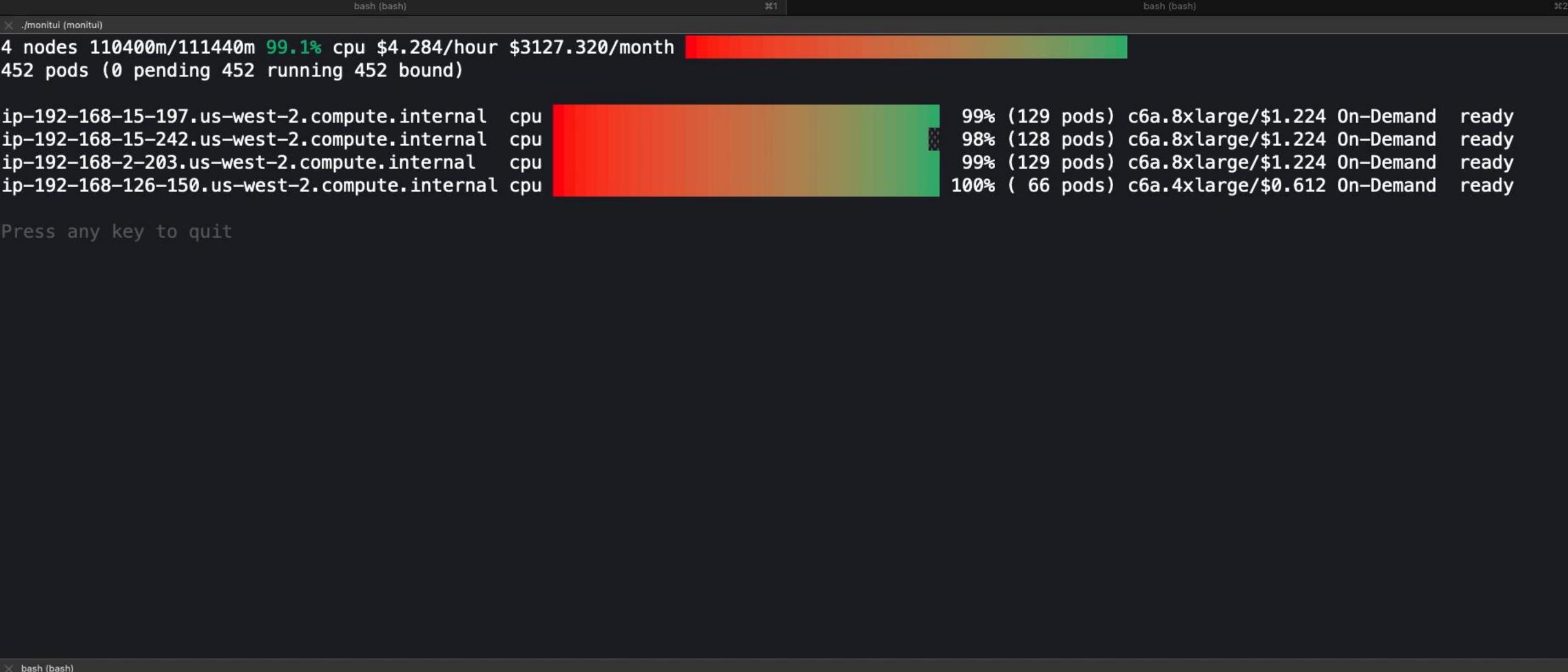
Press any key to quit

bash (bash)

bash-5.1\$ kubectl scale deployment inflate --replicas=600
deployment.apps/inflate scaled
bash-5.1\$

```
./monitui (monitui)          961          bash (vi)          962
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  consolidation:
    enabled: false
  limits:
    resources:
      cpu: 5k
      aws.amazon.com/neuron: 0
      nvidia.com/gpu: 0
      amd.com/gpu: 0
  requirements:
  - key: kubernetes.io/arch
    operator: In
    values:
    - amd64
  - key: "karpenter.sh/capacity-type"
    operator: In
    values:
    - on-demand
  - key: karpenter.k8s.aws/instance-cpu
    operator: Lt
    values: ["33"]
  - key: "topology.kubernetes.io/zone"
    operator: In
    values: ["us-west-2a", "us-west-2b", "us-west-2c", "us-west-2d"]
provider:
  amiFamily: AL2
  apiVersion: extensions.karpenter.sh/v1alpha1
  kind: AWS
  securityGroupSelector:
    karpenter.sh/sg-discovery: eks-demo
  subnetSelector:
    karpenter.sh/sn-discovery: eks-demo
~
```

```
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  consolidation:
    enabled: false
  limits:
    resources:
      cpu: 5k
      aws.amazon.com/neuron: 0
      nvidia.com/gpu: 0
      amd.com/gpu: 0
  requirements:
  - key: kubernetes.io/arch
    operator: In
    values:
    - amd64
  - key: "karpenter.sh/capacity-type"
    operator: In
    values:
    - on-demand
  - key: karpenter.k8s.aws/instance-cpu
    operator: Lt
    values: ["33"]
  - key: "topology.kubernetes.io/zone"
    operator: In
    values: ["us-west-2a", "us-west-2b", "us-west-2c", "us-west-2d"]
provider:
  amiFamily: AL2
  apiVersion: extensions.karpenter.sh/v1alpha1
  kind: AWS
  securityGroupSelector:
    karpenter.sh/sg-discovery: eks-demo
  subnetSelector:
    karpenter.sh/sn-discovery: eks-demo
~  
~  
"provisioner.yaml" 36L, 830B
```



bash (bash)

```
bash-5.1$ kubectl scale deployment inflate --replicas=600
deployment.apps/inflate scaled
bash-5.1$ kubectl scale deployment inflate --replicas=500
deployment.apps/inflate scaled
bash-5.1$ kubectl scale deployment inflate --replicas=440
deployment.apps/inflate scaled
bash-5.1$ 
```



./monitui (monitui) bash (bash)

```
bash-5.1$ kubectl scale deployment inflate --replicas=600
deployment.apps/inflate scaled
bash-5.1$ kubectl scale deployment inflate --replicas=500
deployment.apps/inflate scaled
bash-5.1$ kubectl scale deployment inflate --replicas=440
deployment.apps/inflate scaled
bash-5.1$
```

Public roadmap

- Stay up to date with what we're working on
- Give us feedback and propose ideas
- Get notified when new features ship



github.com/aws/containers-roadmap

Thank you!

Sai Vennam

linkedin.com/in/saivennam



▶ Containers from the Couch