THE ARCHITECTURE OF A MULTI-CLOUD ENVIRONMENT WITH KUBERNETES

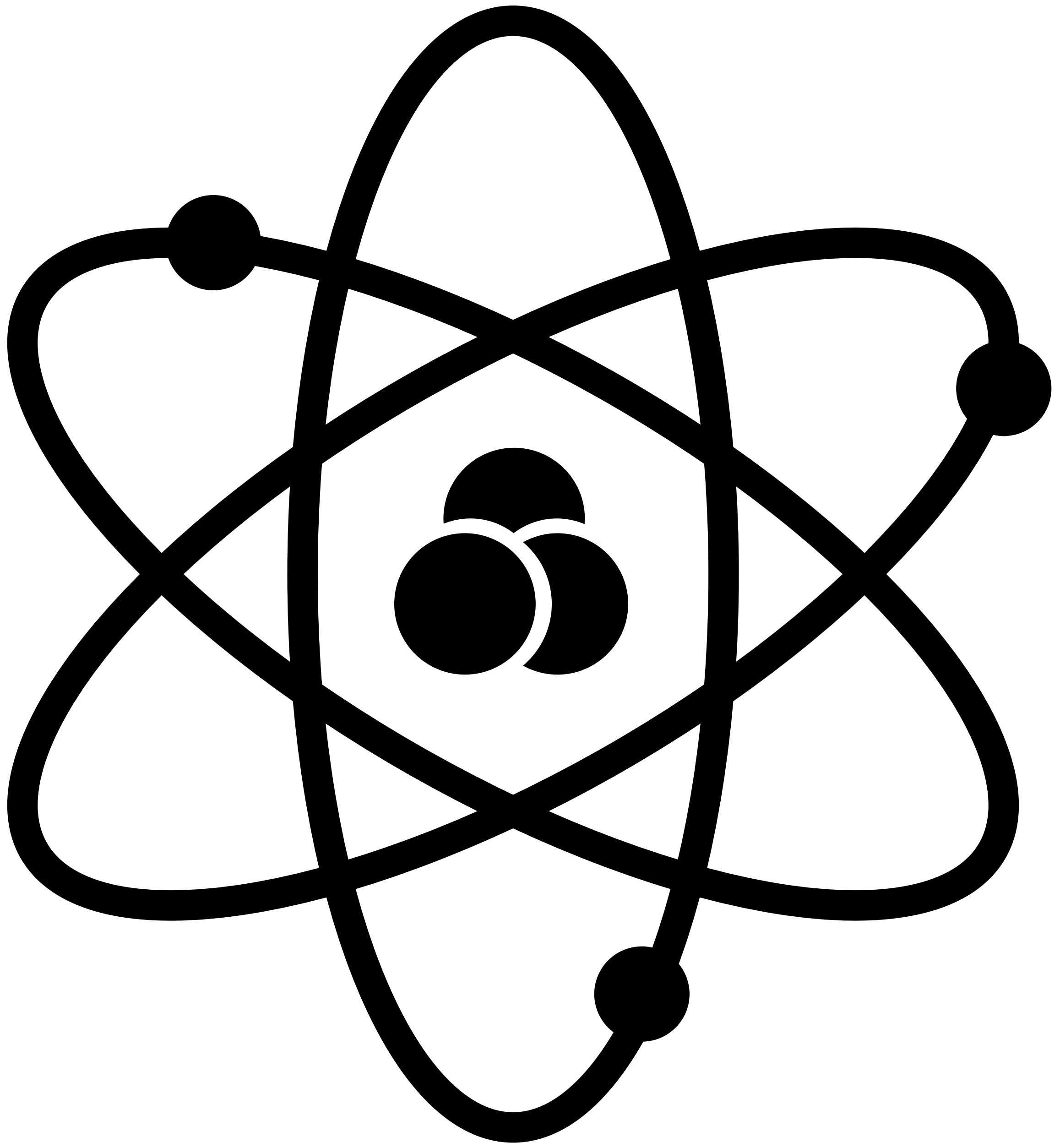
Brian Redbeard

CoreOS

brianredbeard

LET ME TELL YOU SOME

LIES



atom by Jake Schirmer from the Noun Project

?

WHO

- Organizations looking to run Kubernetes in a redundant manner

WHAT

- What to consider when building out a multi-cloud k8s system?
- What are common misunderstandings?

WHEN

- When do these features hit general availability (GA)?

WHERE

- What are the considerations? (broken down by compute environment)

WHY

- Can you answer *why* you want to do this?

HOW

- How do we achieve these goals?

SECTION 1

THE PAST

CLOUD KIDS

THIS TIDBIT IS FOR YOU

"Clouds" by Edward Stojakovic

TO UNDERSTAND WHERE WE'RE AT

LET'S ANALYZE

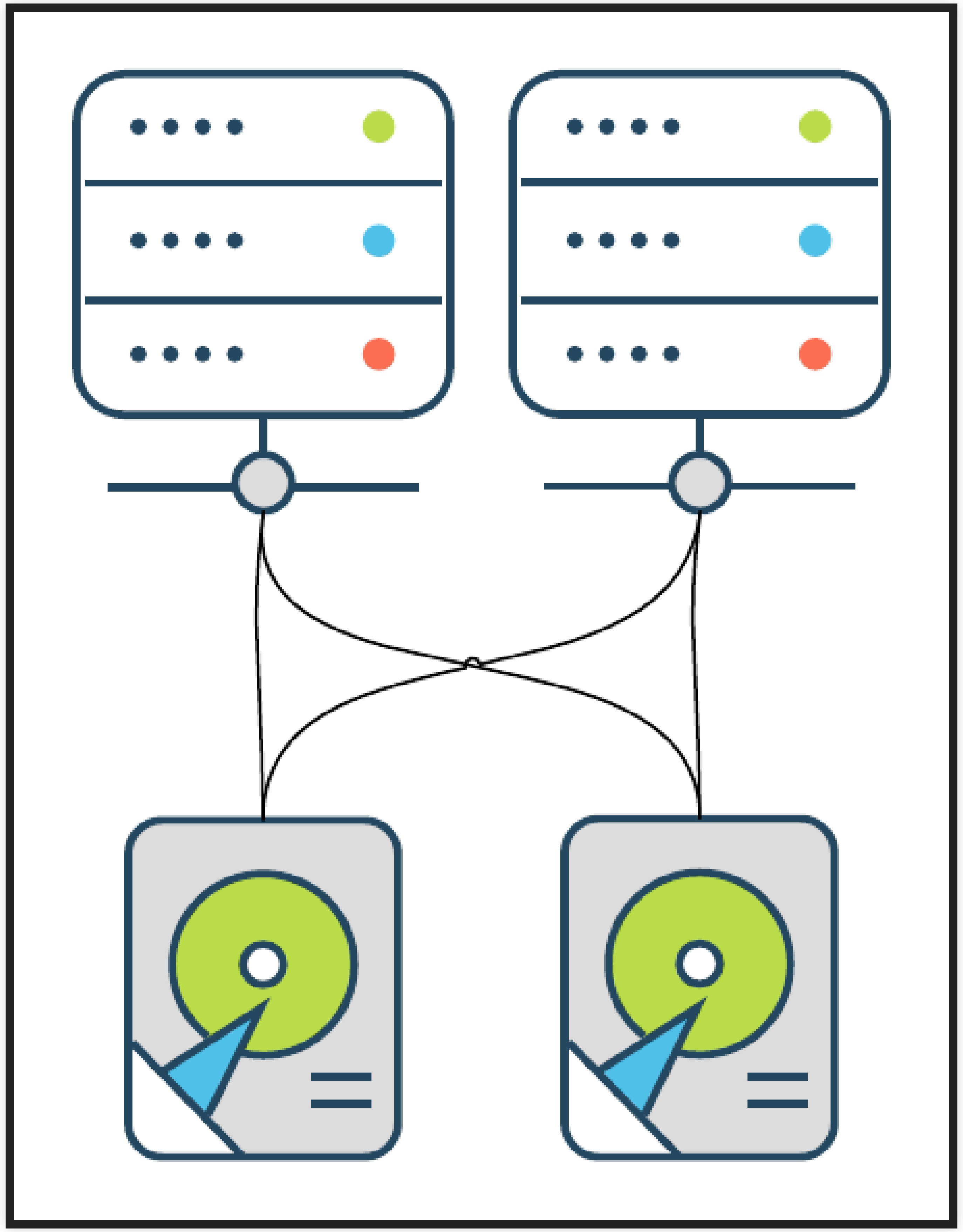
AN EXAMPLE FROM THE PAST

ORACLE RAC

REAL APPLICATION CLUSTERS

TO USE RAC ONE HAD TO HAVE A

SAN



USING A SAN MEANT DEALING WITH

"WWN"S

WELCOME TO THE WONDERFUL WORLD OF

WORLD WIDE NAMES

1000B4744753DB5D

- 10.00.B4.74.47.53.DB.5D
- B4.74.47.53.DB.5D
- B4:74:47:53:DB:5D
- B4:74:47 53:DB:5D

BUT THIS (LACK OF KNOWLEDGE) IS DRIVEN BY

THE CLOUD

DON'T GET BOGGED DOWN BY WHAT THE CLOUD CAN'T DO

SECTION 2

PLANNING

MAPPING OUR NEEDS

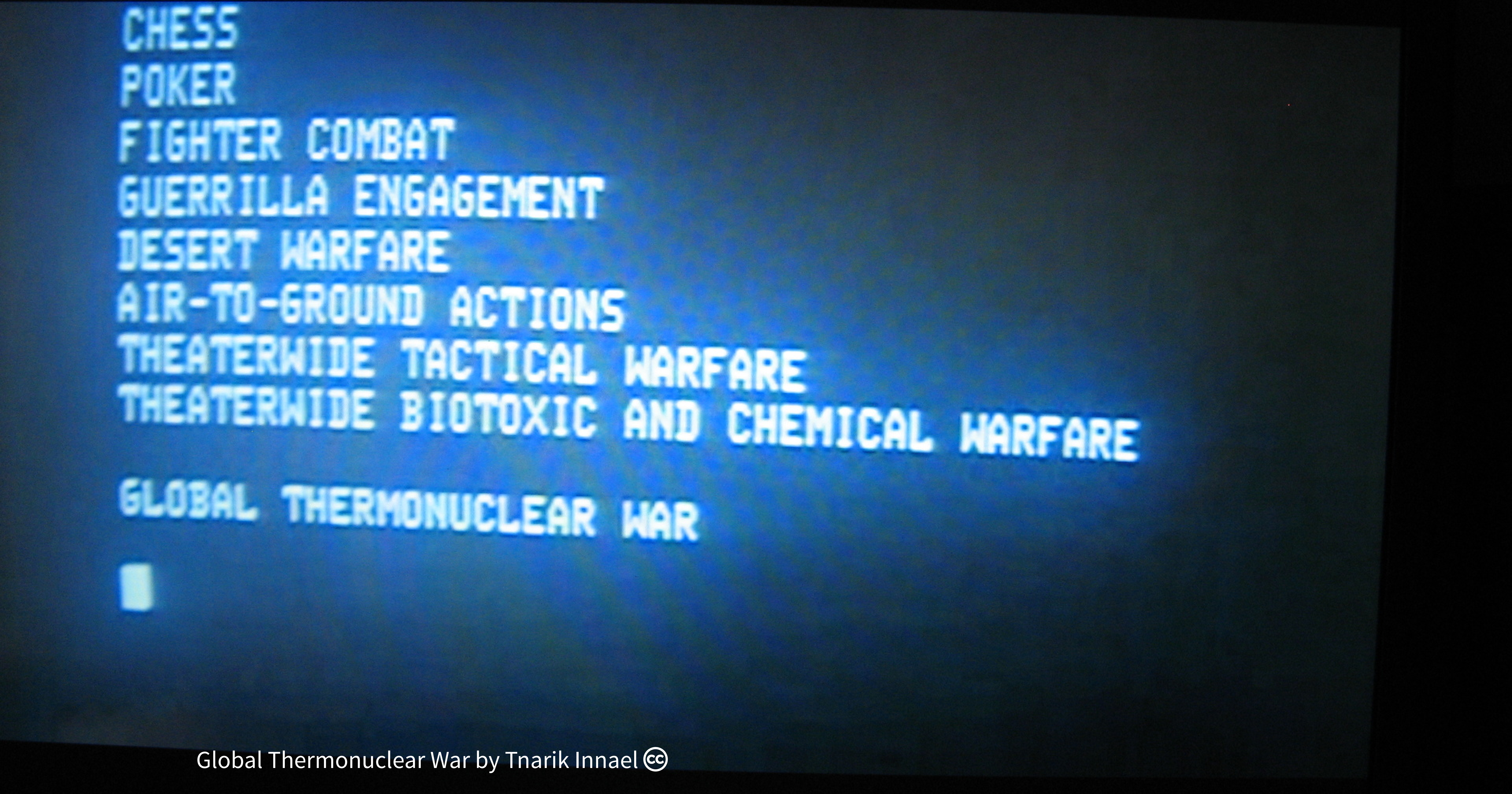
STEP 1

KNOW THE PROBLEM ARE YOU SOLVING FOR

Sarah Shows Everyone What Really Running Is by Charles Barilleaux 🅭

A pile of RAM by Blake Patterson ©

fsociety



```
CHESS
POKER
FIGHTER COMBAT
GUERRILLA ENGAGEMENT
DESERT WARFARE
AIR-TO-GROUND ACTIONS
THEATERWIDE TACTICAL WARFARE
THEATERWIDE BIOTOXIC AND CHEMICAL WARFARE

GLOBAL THERMONUCLEAR WAR

█
```

AKA

DO YOU KNOW WHAT YOUR FAILURE DOMAINS ARE?

ARE YOU PREPARED TO ANSWER THESE

?

HINT: THE ANSWER SHOULD BE YES

STEP 2

DEFINE YOUR ENVIRONMENTS VIA CONFIGURATION MANIFESTS

AKA

CONFIGURATION AS CODE

HOW DOES REDBEARD DO IT?

GIT

OBJECT STORAGE AND CONTROL

JENKINS

REPO MONITORING AND (RE)ACTION

GIT-CRYPT

GPG BASED STORAGE OF SECRETS

BUT YOU DON'T HAVE TO TAKE MY WORD FOR IT

Reading Rainbow © PBS

- terraform
- kops
- kubespray
- helm charts

THE TOOLING IS LESS IMPORTANT THAN COMMITTING TO THE PROCESS

STEP 3

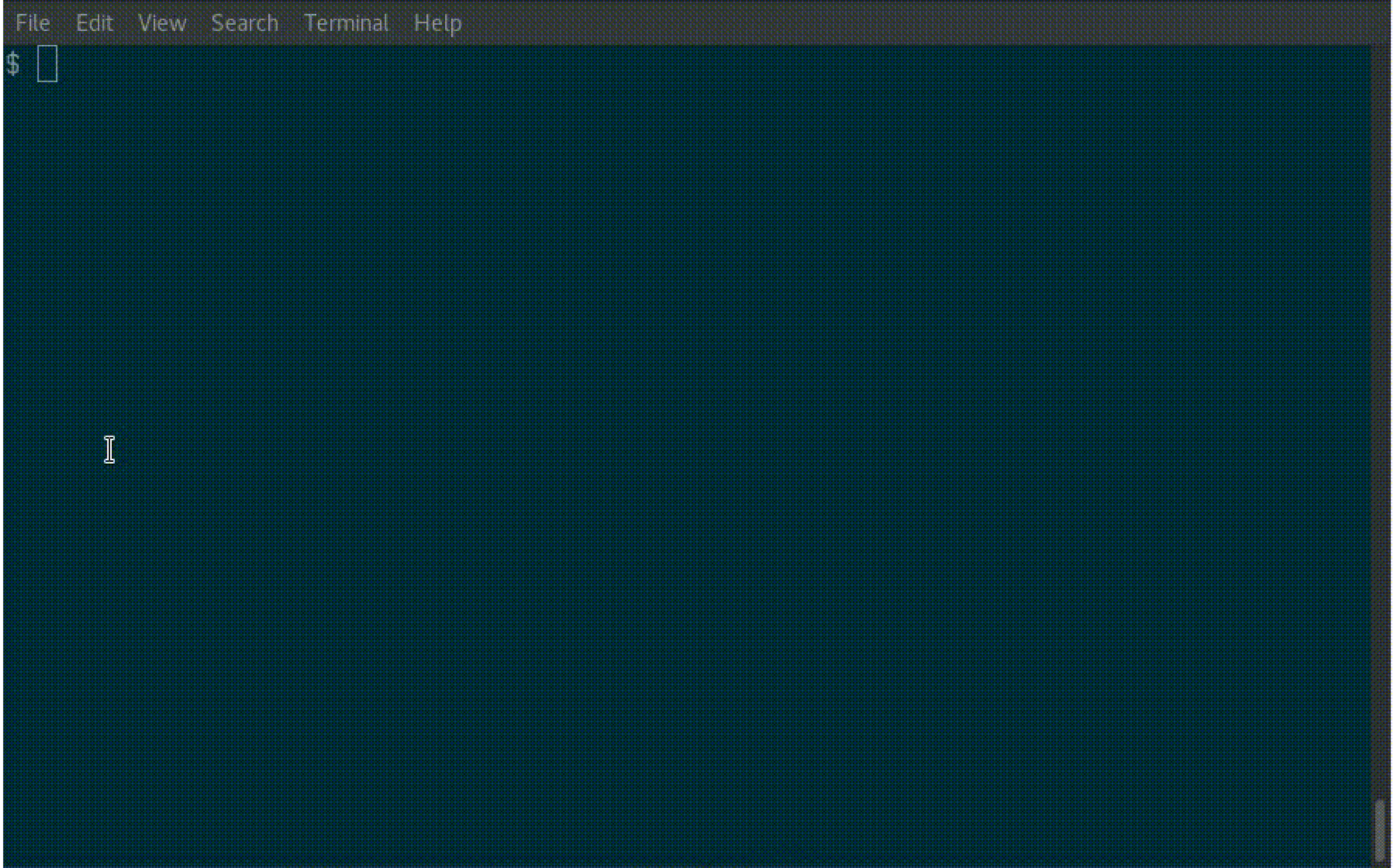
ENSURE THAT YOUR NETWORK RANGES ARE

NON OVERLAPPING

This doesn't seem like a big deal, but please... just make sure the ranges do not overlap.

NEED SOME HELP WITH THIS?

HTTPS://GITHUB.COM/TSCHUY/CIDRBLOCKS


File
Edit
View
Search
Terminal
Help

```
$ 
```

NEED SOME HELP WITH THIS?

HTTPS://GITHUB.COM/TSCHUY/CIDRBLOCKS

STEP 4

KNOW WHAT YOUR STORAGE IS

STOP FREAKING OUT ABOUT STORAGE

SERIOUSLY

WHAT'S YOUR PLATFORM?

AWS

- Elastic Block Storage (EBS)

GCP

- Persistent Disks

AZURE

- Disks
  - (Premium / Standard)
  - (Managed / Unmanaged)

WHAT'S (GENERICALLY) HAPPENING UNDER THE HOOD?

KUBERNETES IS GIVING YOU SHOULDERS TO STAND ON

RWO

`ReadWriteOnce`

ROX

`ReadOnlyMany`

RWX

`ReadWriteMany`

JUST BECAUSE YOU'VE NEVER DONE IT ON THE CLOUD

DOESN'T MEAN IT'S NOT POSSIBLE

BARE METAL

HINT: IT WORKS VERY SIMILARLY

- SAN Disks (iSCSI, Fibre Channel, etc)
- Cinder
- Ceph (cephfs / RBD)
- etc

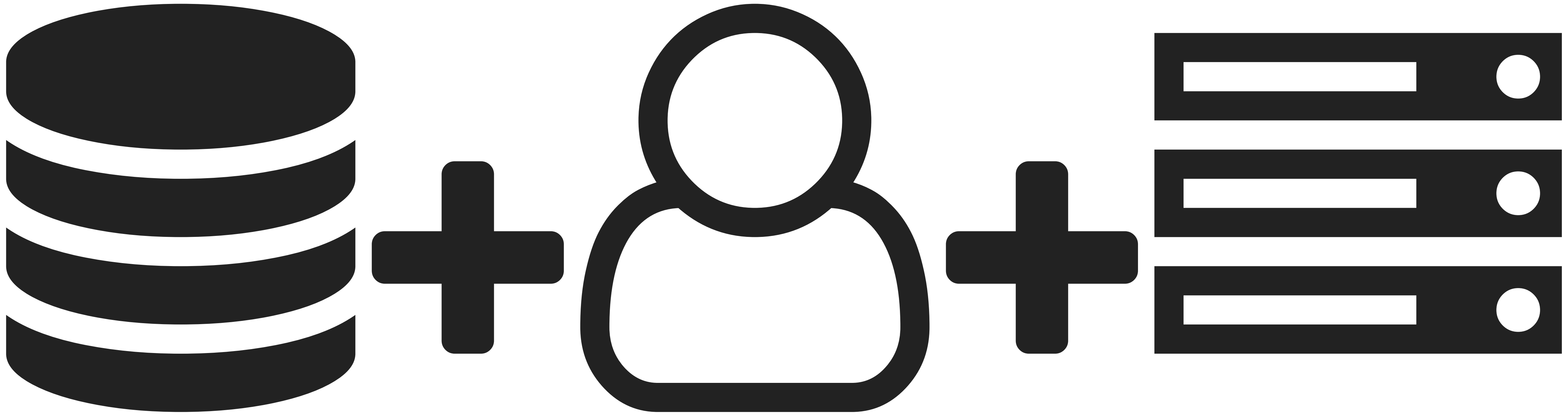
YOU JUST NEED AN API FOR STORAGE

APIS FOR STORAGE EXIST

AND IT DOESN'T HAVE TO BE `SPENSIVE

FREENAS.ORG

free/libre IP SAN FreeBSD distro



CONFIGURE RAID

EXPORT ISCSI

SECTION 3

SETUP

PUTTING TOGETHER THE PIECES

Lego Advent 2015 Day 15 by Bill Ward 🅭

SO WHAT DO WE NEED TO WORRY ABOUT?

HINT: THEY'RE THE THINGS YOU SHOULD ALREADY BE DOING

SINGLE SIGN ON

DON'T RUN LOCAL USERS ON YOU CLUSTERS

components
dex
ldap / oidc

LOG AGGREGATION

COLLECT LOGS (HOST, K8S, APPLICATION) IN A CENTRAL LOCATION

components
fluentd
elasticsearch
kibana

MONITORING & ALERTING

MEASURE PERFORMANCE & ALERT ON PROBLEMS

components
prometheus (metrics)
alert manager (alerting)
jaeger

DNS CONFIGURATION

FEDERATE YOUR DNS

components
coredns
your existing DNS infrastructure

RBAC CONFIGURATION

SYNC YOUR CONFIGS ACROSS YOUR CLUSTERS

components

---

continuous deployment (jenkins, spinnaker)

TRAFFIC DISTRIBUTION

CLUSTER TRAFFIC NEEDS REDUNDANCY

components
cloud load balancers
BGP + ECMP
F5 / Netscaler / ACE

DEMAND NETWORK APIS

IN DISTRIBUTED SYSTEMS, EVERYTHING IS A NETWORK SERVICE

SECTION 4

EXECUTION

TAKING ACTION

STEP 1

CLUSTER PROVISIONING

UNDERSTAND THE STAGES OF CLUSTER INITIALIZATION

- host deployment & configuration
  - etcd deployment
  - master node deployment
  - worker node deployment
- cluster configuration

HOST DEPLOYMENT & CONFIGURATION

- Normalize & templatize your host configuration (Easy with Container Linux)
  - same manifests can be used for bare metal and cloud
  - If using kickstart + cloud-config break things down to minimal state (or use ansible)
- Avoid "static" configs (network, etc)

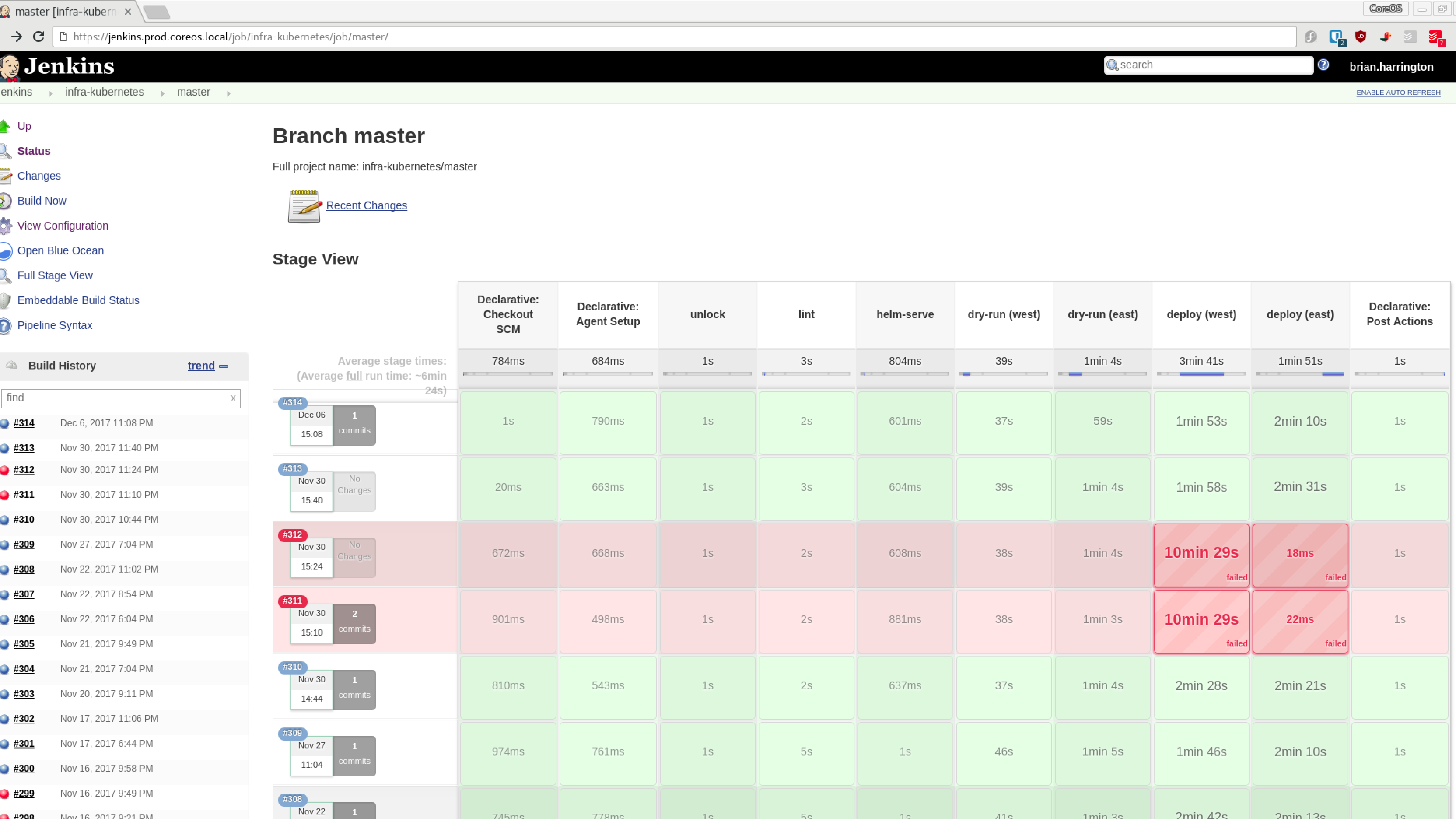
CLUSTER CONFIGURATION

- Kubelet flags - Ensure everything is "under management"
- Use robots* to do your bidding

ROBOTS YOU SAY?!

PEOPLE LOVE TO HATE ON JENKINS...

BUT THIS BUTLER DOES OUR BIDDING


master [infra-kubern ×

https://jenkins.prod.coreos.local/job/infra-kubernetes/job/master/

Jenkins — search — brian.harrington

Jenkins › infra-kubernetes › master › ENABLE AUTO REFRESH

- Up
- Status
- Changes
- Build Now
- View Configuration
- Open Blue Ocean
- Full Stage View
- Embeddable Build Status
- Pipeline Syntax

Build History trend

find

#314	Dec 6, 2017 11:08 PM
#313	Nov 30, 2017 11:40 PM
#312	Nov 30, 2017 11:24 PM
#311	Nov 30, 2017 11:10 PM
#310	Nov 30, 2017 10:44 PM
#309	Nov 27, 2017 7:04 PM
#308	Nov 22, 2017 11:02 PM
#307	Nov 22, 2017 8:54 PM
#306	Nov 22, 2017 6:04 PM
#305	Nov 21, 2017 9:49 PM
#304	Nov 21, 2017 7:04 PM
#303	Nov 20, 2017 9:11 PM
#302	Nov 17, 2017 11:06 PM
#301	Nov 17, 2017 6:44 PM
#300	Nov 16, 2017 9:58 PM
#299	Nov 16, 2017 9:49 PM
#298	Nov 16, 2017 9:21 PM

Branch master

Full project name: infra-kubernetes/master

Recent Changes

Stage View

	Declarative: Checkout SCM	Declarative: Agent Setup	unlock	lint	helm-serve	dry-run (west)	dry-run (east)	deploy (west)	deploy (east)	Declarative: Post Actions
Average stage times: (Average full run time: ~6min 24s)	784ms	684ms	1s	3s	804ms	39s	1min 4s	3min 41s	1min 51s	1s
#314 Dec 06 15:08 — 1 commits	1s	790ms	1s	2s	601ms	37s	59s	1min 53s	2min 10s	1s
#313 Nov 30 15:40 — No Changes	20ms	663ms	1s	3s	604ms	39s	1min 4s	1min 58s	2min 31s	1s
#312 Nov 30 15:24 — No Changes	672ms	668ms	1s	2s	608ms	38s	1min 4s	10min 29s failed	18ms failed	1s
#311 Nov 30 15:10 — 2 commits	901ms	498ms	1s	2s	881ms	38s	1min 3s	10min 29s failed	22ms failed	1s
#310 Nov 30 14:44 — 1 commits	810ms	543ms	1s	2s	637ms	37s	1min 4s	2min 28s	2min 21s	1s
#309 Nov 27 11:04 — 1 commits	974ms	761ms	1s	5s	1s	46s	1min 5s	1min 46s	2min 10s	1s
#308 Nov 22 — 1	745ms	778ms	1s	5s	1s	41s	1min 3s	2min 42s	2min 13s	1s

K8S DEPLOYMENTS

step by step:

PIPELINE STAGES:

- unlock credentials (`git-crypt`)

PULL OUR KEY FROM ESCROW (PT 1)

```
def gitCryptUnlock(credsId) {
  def key = [file(credentialsId: credsId, variable: 'GIT_CRYPT_K

  withCredentials(key) {
    ansiColor('xterm') {
      sh '''#!/bin/bash -xe
            git-crypt status >/dev/null
            git-crypt unlock <(base64 -d "${GIT_CRYPT_KEY}")
         '''
    }
  }
}
```

PULL OUR KEY FROM ESCROW (PT 2)

```
  stage('unlock') {
    steps {
      script {
        gitCryptUnlock('infra-terraform-key')
      }
    }
```

PIPELINE STAGES:

- unlock credentials (`git-crypt`)
- lint/validate the config (`git clean -fdx` & `terraform validate`)

LINT AND VALIDATE OUR CONFIG (PT 1)

```
stage('validate') {
  steps {
    withCredentials(aws['coreos']) {
      ansiColor('xterm') {
          sh 'git clean -fdx'
          sh './scripts/ci'
      }
    }
  }
}
```

LINT AND VALIDATE OUR CONFIG (PT 2)

```
LINT_DIRS=(global us-west-1 us-central1)
for ldir in ${LINT_DIRS[@]}; do
    for dir in $(find "${ldir}" -name 'backend.tf' -printf \
      '%h\n' | uniq); do
        pushd "${dir}" >/dev/null

        # This is required as of Terraform 0.10.0 because
        # the plugins must be downloaded before validation.
        terraform init -input=false

        terraform validate
        popd
    done
done
```

LINT AND VALIDATE OUR CONFIG (PT 3)

```
for ldir in ${LINT_DIRS[@]}; do
  # The `fmt` command doesn't seem to exit non-zero if there
  # are formatting changes needed.
  FILES="$(terraform fmt -list=true -write=false "${ldir}")"

  if [ -n "${FILES}" ]; then
      echo "==> The following files need formatting changes:"
      echo "${FILES}"
      exit 1
  fi
done
```

PIPELINE STAGES:

- unlock credentials (`git-crypt`)
- lint/validate the config
  - `git clean -fdx`
  - `terraform validate`
- do a dry run (`terraform plan`)
- ask for a human to confirm the work (slack message)
- deploy (`terraform apply`)

WORKING IN THIS WAY, ADDING A NEW CLUSTER IS AS EASY AS DEFINING AN ENVIRONMENT

STEP 2

Maintaining what you have

WE ALSO USE A SIMILAR SET OF JOBS TO *MANAGE* THE EXISTING ENVIRONMENTS

ENVIRONMENT DEFINITIONS:

```
def clusters = [
 'us-west-1': [
   ['name': 'dev-v1519',  'path': 'clusters/us-west-1/dev-v1519.
   ['name': 'prod-v1472', 'path': 'clusters/us-west-1/prod-v1472
 ],

 'us-east-1': [
   ['name': 'prod-v1507', 'path': 'clusters/us-east-1/prod-v1507
  ]
]
```

(PRETTY FAMILIAR) PIPELINE STAGES:

- unlock credentials (`git-crypt`)
- lint/validate the config
  - `git clean -fdx`
  - `helm lint`
- helm upgrade

THANKS TO BRAD ISON WITH COREOS INFRASTRUCTURE

□BISON

STEP 3

Manual operations

DON'T

JUST DON'T

FINE...

MAKE SURE YOU

--EXPORT

```
kubectl get -o yaml --export=true deployment myapp
```

Thanks to Duffie Cooley! - coresolve

THIS REMOVES ALL CLUSTER SPECIFIC INFORMATION FROM THE RESOURCE

You're then just a `kubectl apply` *away from mucking up all of the work your robot has done*

STEP 4

Disaster recovery

THINGS WILL GO WRONG

HAVE A PLAN

TEST THE PLAN

BACKUP ETCD

```
ETCDCTL_API=3 /opt/bin/etcdctl snapshot save backup.db
```

RUN MULTIPLE CLUSTERS IN *DIFFERENT* FAILURE DOMAINS

USE PURPOSE BUILT TOOLS

/HEPTIO/ARK

"A UTILITY FOR MANAGING DISASTER RECOVERY, SPECIFICALLY FOR YOUR KUBERNETES CLUSTER RESOURCES AND PERSISTENT VOLUMES"

WHO

- You

WHAT

- What to consider when building out a multi-cloud and multi-environment k8s system?
- What are common misunderstandings?

WHEN

- When do these features hit general availability (GA)?
    - All of this is possible **today**

WHERE

- What are the considerations? (broken down by compute environment)

WHY

- Can you answer *why* you want to do this?

HOW

- How do we achieve these goals?

TECTONIC

by CoreOS

https://coreos.com/tectonic

Free for up to 10 nodes!

WE'RE HIRING

CoreOS

https://coreos.com/jobs

THANKS

brianredbeard

brianredbeard

redbeard@coreos.com