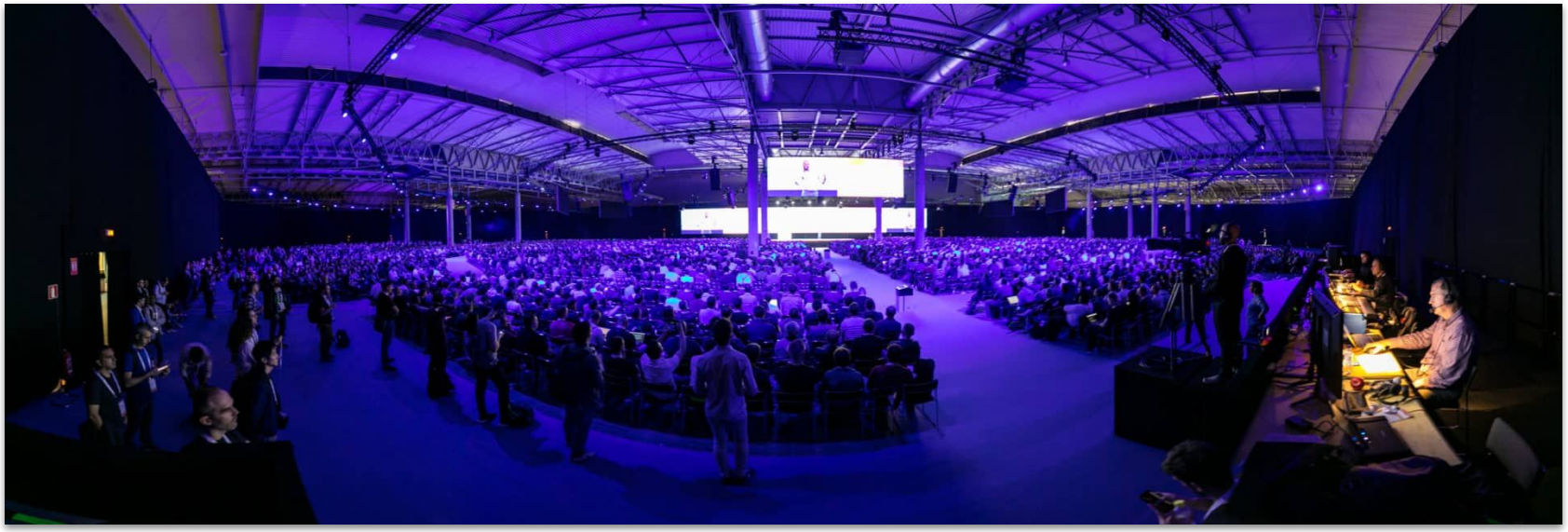
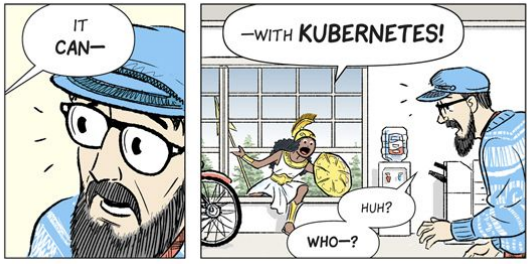
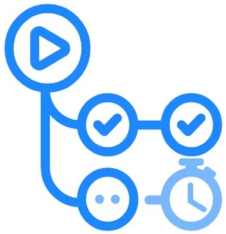
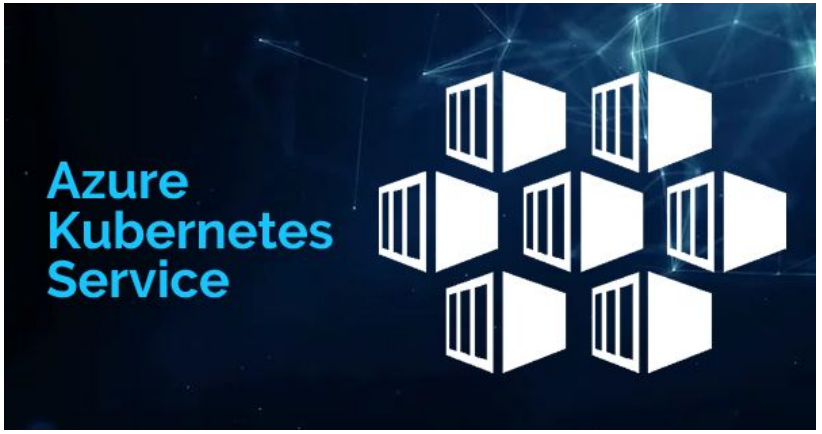
AKS LifeCycle withGitHub Action

(Praparn Lueangphoonlap)









kubernetes by Google

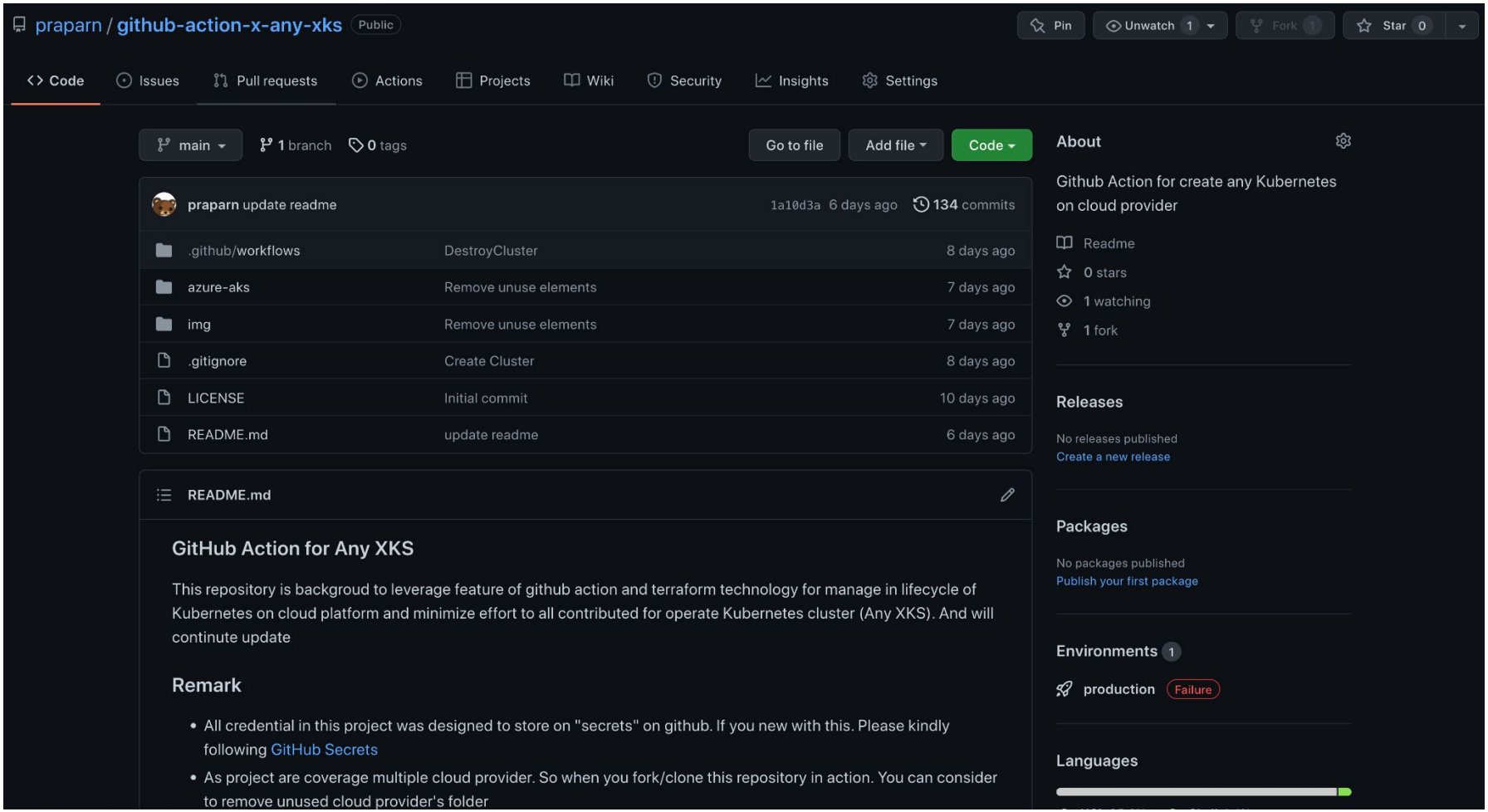
Agenda

- Rising of cloud-manage for Kubernetes
- GitHub Action for any XKS
  - Purpose of Project
  - Architecture Design
- AKS Cluster
  - Credential as prerequisites
  - Feature on project
- Demo session
- Q&A

kubernetes by Google

Git Resource

- https://github.com/praparn/github-action-x-any-xks



K8S Lifecycle with Github Action

kubernetes by Google

Reference

- Azure AKS Cluster
- AWS EKS Cluster
- Google GKE Cluster
- Huawei Cloud Container Service
- GitHub Action with Terraform
- Terraform AKS
- Etc.

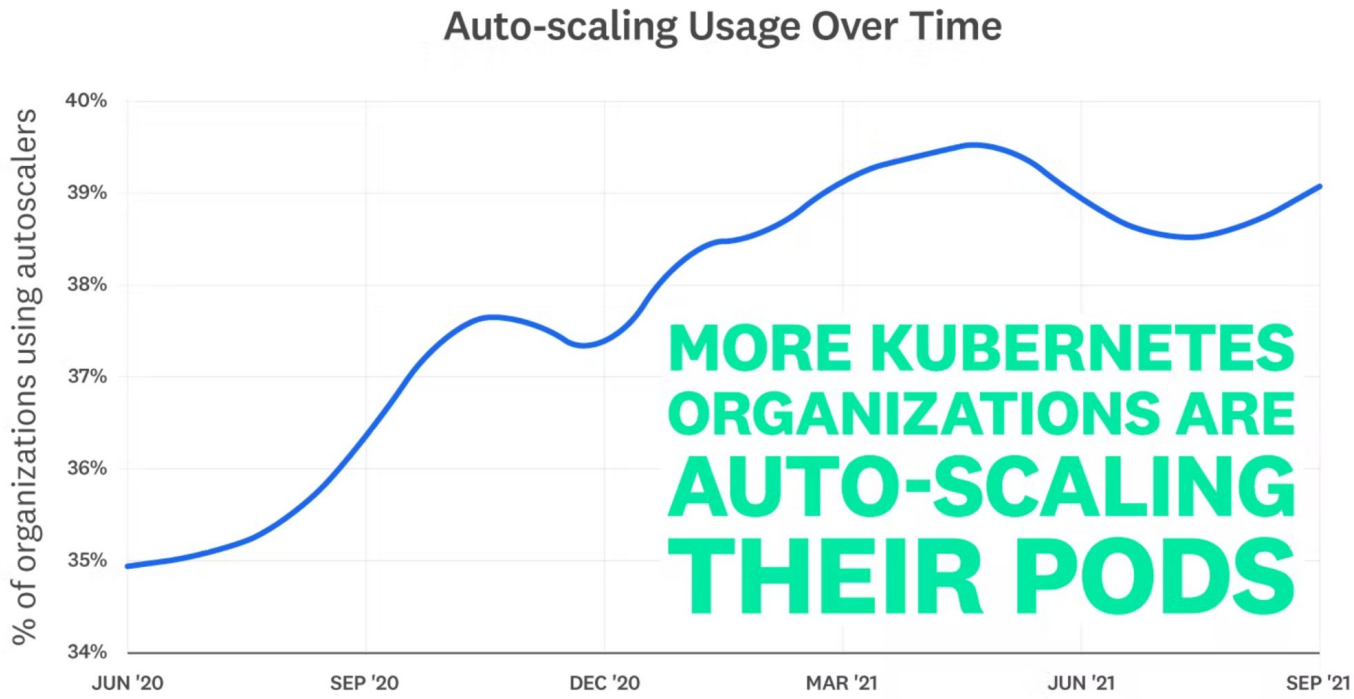
kubernetes by Google

Rising of cloud-manage for Kubernetes

kubernetes by Google

DATADOG

10 TRENDS IN REAL-WORLD CONTAINER USE



Source: Datadog

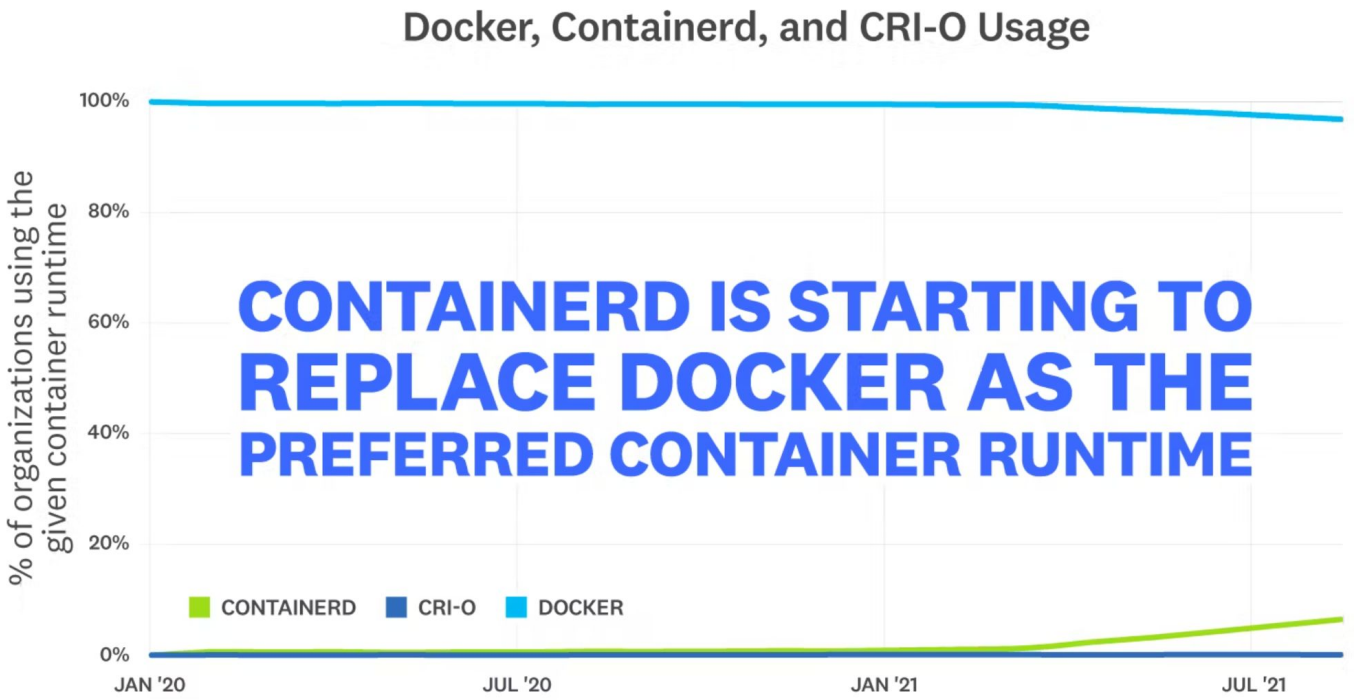
Ref: https://www.datadoghq.com/container-report/

K8S Lifecycle with Github Action

kubernetes by Google

DATADOG

10 TRENDS IN REAL-WORLD CONTAINER USE



Source: Datadog

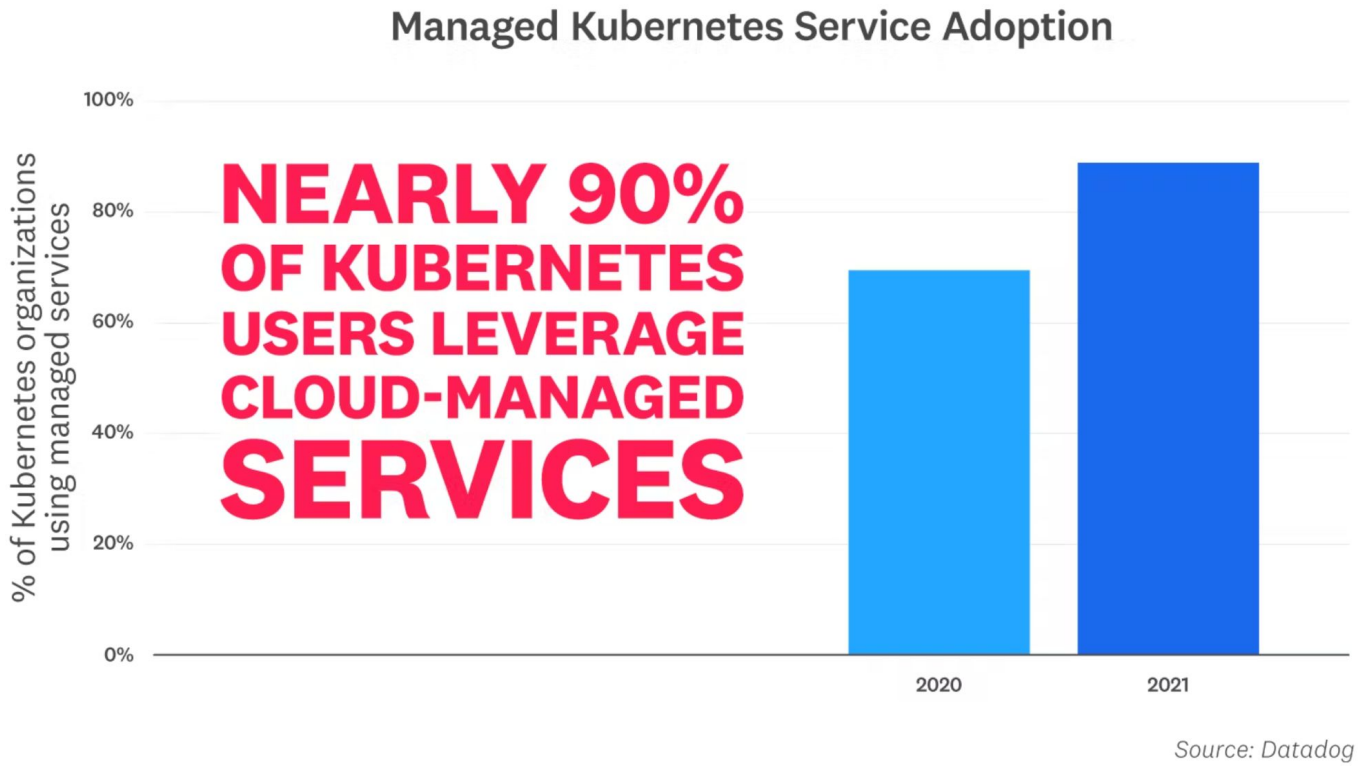
Ref: https://www.datadoghq.com/container-report/

K8S Lifecycle with Github Action

kubernetes by Google

DATADOG

10 TRENDS IN REAL-WORLD CONTAINER USE



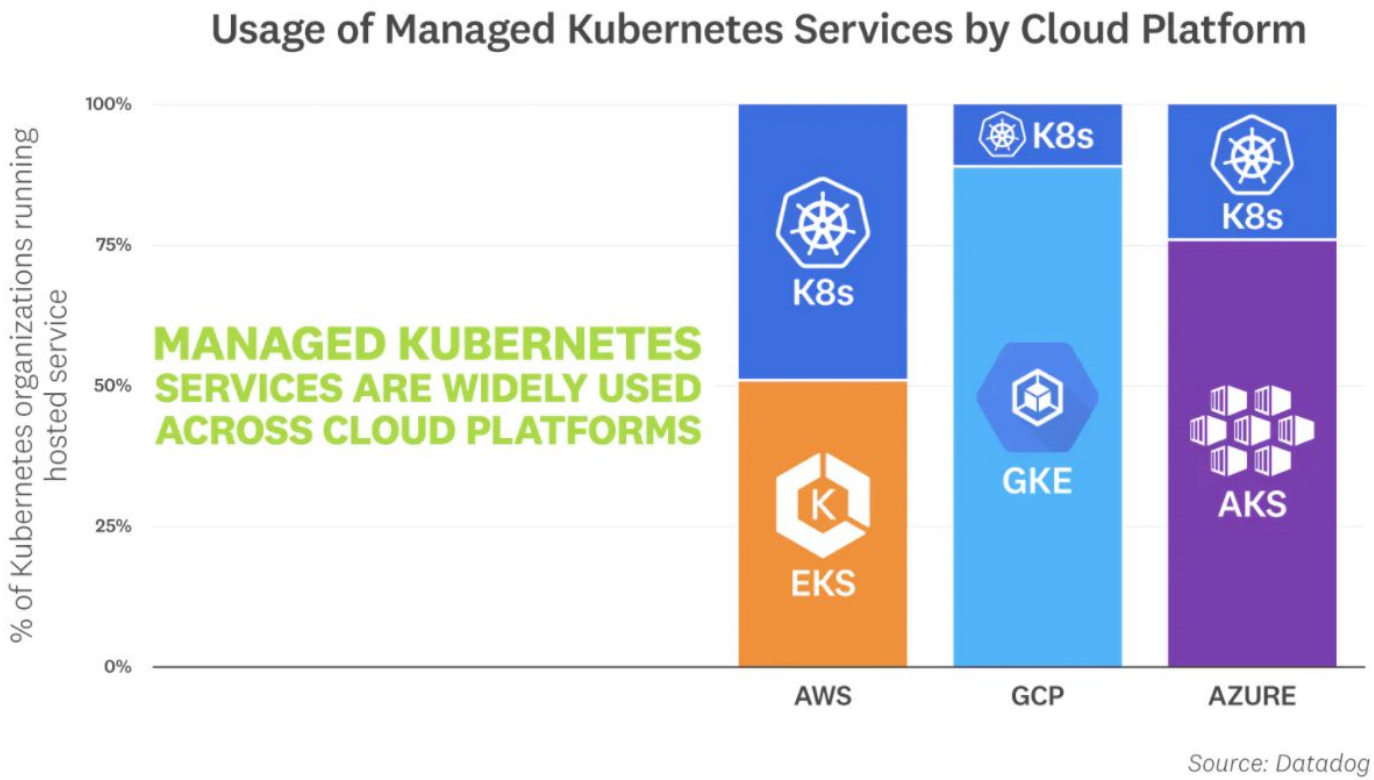
Ref: https://www.datadoghq.com/container-report/

K8S Lifecycle with Github Action

kubernetes by Google

DATADOG

10 TRENDS IN REAL-WORLD CONTAINER USE



Ref: https://www.datadoghq.com/container-report/

K8S Lifecycle with Github Action

kubernetes by Google

GitHub Action for Any XKS

kubernetes by Google

Purpose of Project

- Kubernetes on cloud provider (Kubernetes as service) is on every where.
- Each cloud provider have specific architecture and solution for manage their cloud provider
- This make complicate for implementor /developer/ devops/etc for handle multiple skill-set and lifecycle…
  - Create/Destroy cluster on cloud environment
  - Hardening / Tuning cluster
    - Upgrade Kubernetes version
    - Network plugin/policy
    - Network range for pods/cluster
    - Sysctl
  - Integrated with facilities on cloud
  - Scaling workload (Increase/Decrease/Autoscale etc)
  - etc.

Purpose of Project

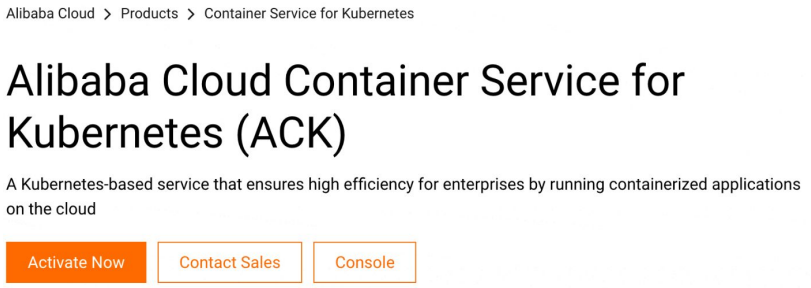
- For our project have main purpose for reduce this complicate for each cloud provider and make all contribute to leverage any "**XKS**" with simple standard
- Project is integrated with "**Terraform**" framework for operate IaC (infrastructure as code) and make same standard on project
- For make it automation part. To make this project help for provision kubernetes cluster automatic. We choose "**Github Action**" as build-in on repository. So this will effort less for operation
- All credential was keep in "**GitHub Secret**" and run via github action. So we not leak any credential to outside

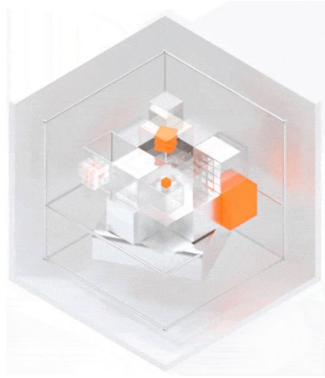
Purpose of Project

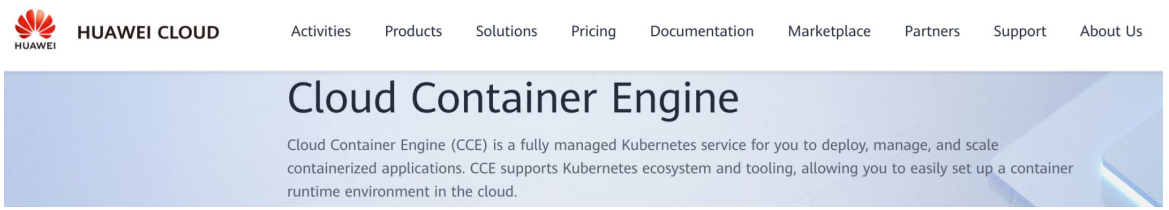
- Project will target for handle all Kubernetes platform in cloud provider

Azure Kubernetes Service (AKS)

Google Kubernetes Engine







K8S Lifecycle with Github Action

kubernetes by Google

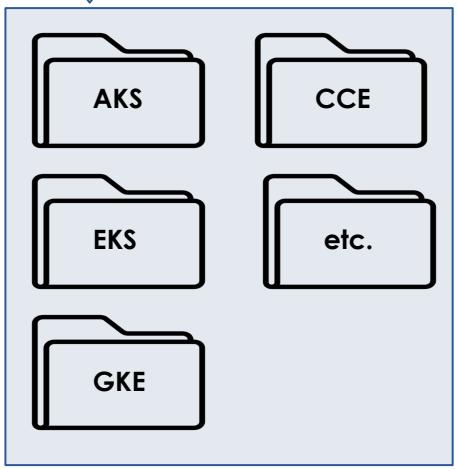
Purpose of Project

- Contributor can clone repository (to private repository) and configure properties that need (include credential). After that just push code to your repository and tag “xxxx” and… done !!!

Purpose of Project

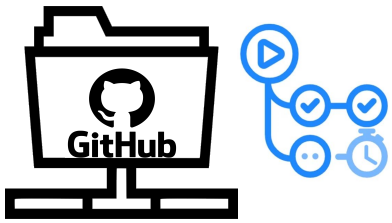
https://github.com/praparn/github-action-x-any-xks
Master Repo

Step1: Git clone (private repository)

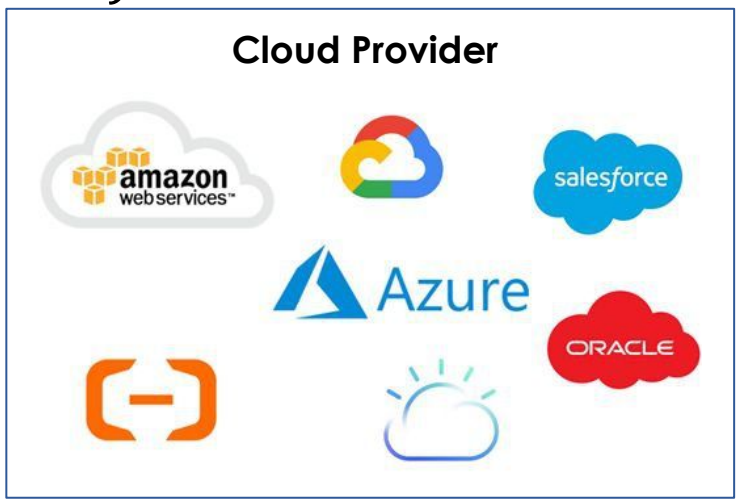


Step2: Configure properties and Credential

Step3: Commit and Push with specific "Tags"

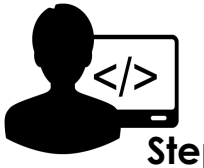


Step4: Github Action will run terraform to create K8S



Step5: Cluster was created

kubernetes



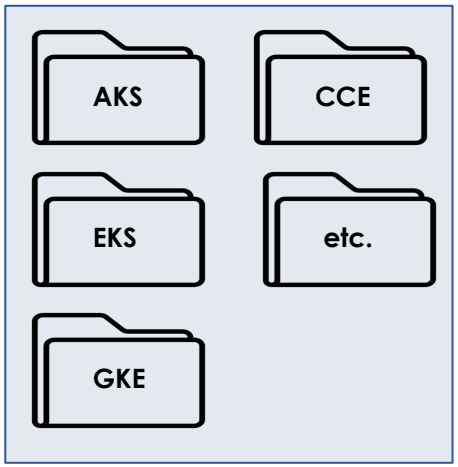
Developer/ Application Owner

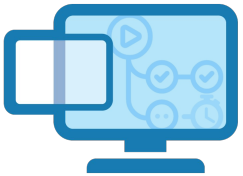
Step6: Access cluster and operate

K8S Lifecycle with Github Action

kubernetes by Google

Architecture Design

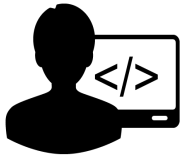




Step1: Create space for housing terraform "state file"
Step2: Terraform was configure remote state file as create
Step3: Run terraform for create/modify/delete cluster as design

kubernetes

GitHub Secret
<credential>.....



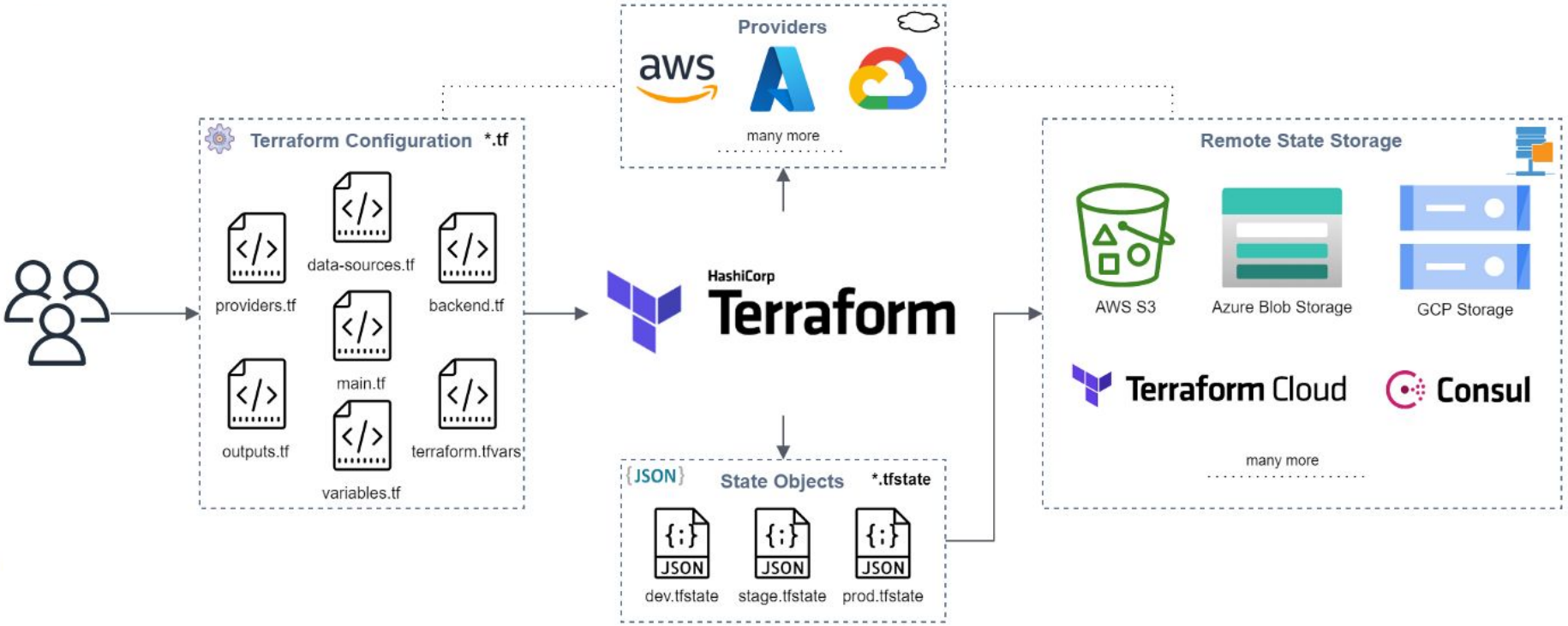
Runner will active with "Tag"

- "xxx-init-env**" (create state file location)
- "xxx-cluster-create**" (create cluster)
- "xxx-cluster-modify**" (modify cluster)
- "xxx-cluster-destroy**" (destroy cluster)
- "xxx-destroy-env**" (destroy state file location)

K8S Lifecycle with Github Action

kubernetes by Google

Architecture Design



Ref:https://medium.com/devops-mojo/terraform-remote-states-overview-what-is-terraform-remote-state-storage-introduction-936223a0e9d0

kubernetes by Google

AKS Cluster

kubernetes by Google

Credential as prerequisites

- {AZURE_CREDENTIALS}: Store output in JSON format of your service principle. If you not yet to create service principal. Please follow this KB Azure Service Principal

```
{
  "clientId": "<GUID>",
  "clientSecret": "<GUID>",
  "subscriptionId": "<GUID>",
  "tenantId": "<GUID>",
  (...)
}
```

- {AZURE_CLIENT_ID}: Input client id (You can check this from "{AZURE_CREDENTIALS}")

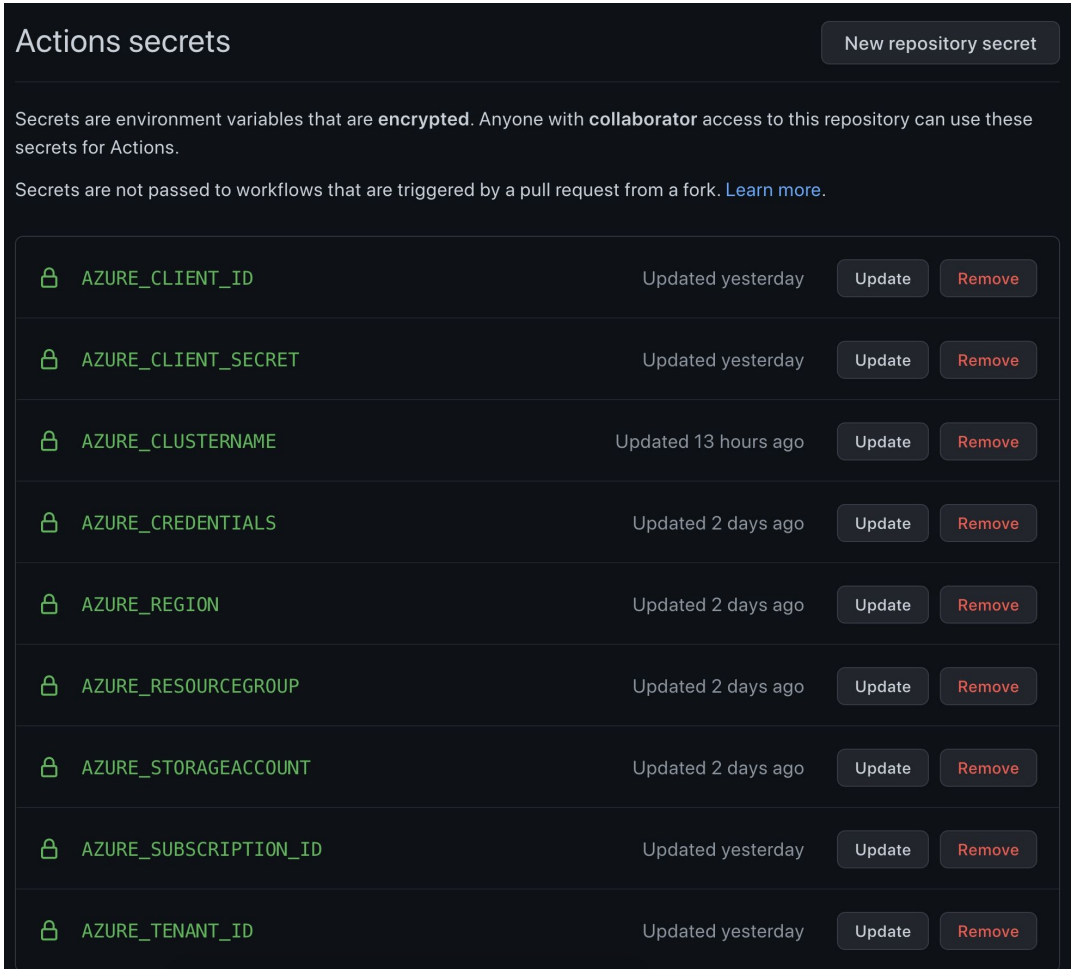
- {AZURE_CLIENT_SECRET}: Input client secret id (You can check this from "{AZURE_CREDENTIALS}")

Credential as prerequisites

- {AZURE_SUBSCRIPTION_ID}: Input subscription id (You can check this from "{AZURE_CREDENTIALS}")
- {AZURE_TENANT_ID}: Input tanant id (You can check this from "{AZURE_CREDENTIALS}")
- {AZURE_REGION}: Input your region on portal. Ex:"eastasia" Region code
- {AZURE_RESOURCEGROUP}: Input your resource group name for create other elements
- {AZURE_STORAGEACCOUNT}: Input your storage account name for keep terraform state on portal. Remark: Storage account name must be between 3 and 24 characters in length and use numbers and lower-case letters only

Credential as prerequisites

- {AZURE_CLUSTERNAME}: Input your AKS cluster name



kubernetes by Google

Feature on Project

- **Full life-cycle** for AKS cluster with gitHub action
- **Init-Environment**:
  - Create "resource group" for reference of all element in Azure portal
  - Create "storage account" for housing file
  - Create "tfstate" on storage account
- **Init-Cluster:**
  - Create "Log analytic"
  - Create "Virtual network" and "Subnet"
  - Create "EKS" cluster with custom configuration
  - Create "Ingress Application Gateway" for application
  - Export credential of Kubernetes to file "aks-config" and commit back to git repository
  - Deploy demo application for test cluster

Feature on Project

- **Modify-Cluster:**
  - Upgrade Kubernetes version
  - Increase/Decrease worker node
  - Cluster autoscaling feature
  - Custom configuration
  - etc.
- **Destroy-Cluster**:
  - Delete AKS and resource related

- **Destroy-Environment:**
  - Delete tfstate and blob storage
  - Delete resource group

Feature on Project (Custom Config)

```
##############################
#      General Properties    #
##############################
variable region {
   default     = "eastasia"
   description = "Region for housing EKS (https://azuretracks.com/2021/04/current-azure-region-names-reference/)"
}

variable location {
   default = "East Asia"
  description = "Full name of region for housing EKS (https://azuretracks.com/2021/04/current-azure-region-names-reference/)"
}

variable environment {
   default = "Development"
   description = "Environment to deploy (Development, UAT, Production)"
}

variable ssh_public_key {
   default = "~/.ssh/id_rsa.pub"
   description = "SSH public key location"
}

variable network_space {
   default = "10.255.0.0/16"
   description = "Total network cidr for cluster"
}

variable network_space_aks {
   default = "10.255.0.0/20"
   description = "Total network cidr for AKS cluster itself (default: 10.255.0.0 - 10.255.15.254 at 4094 hosts)"
}

variable network_space_aksapp {
   default = "10.255.16.0/20"
   description = "Total network cidr for AKS cluster itself (default: 10.255.16.0 - 10.255.31.254 at 4094 hosts)"
}
```


kubernetes by Google

Feature on Project (Custom Config)

```
##############################
#      AKS Properties        #
##############################

variable cluster_public {
  default = true
  description = "AKS cluster enable access from public or not (true/false)"
}


variable default_node_pool {
  default = "aksnodepool"
  description = "AKS node pool name (Avoid '-')"
}

variable cluster_version {
  default = "1.21.9"
  description = "AKS K8S version (1.21.9/1.22.4/1.22.6)"
}

variable cluster_autoscale {
  default = false
  description = "AKS autoscale"
}

variable cluster_autoscale_max {
  default = null
  description = "AKS maximum autoscale (If not autoscale. This need to be null. (1 - 1000)"
}

variable cluster_autoscale_min {
  default = null
  description = "AKS minimum autoscale (If not autoscale. This need to be null). (1 - 1000)"
}

variable aks_zone {
  default = ["1", "2", "3"]
  description = "AKS avaliable on which zone of Azure Cloud"
}
```

```
variable cluster_httprouting {
  default = false
  description = "AKS enable http routing (true/false). (Recommend enable on DEV environment)"
}

variable "worker_count" {
  default = 3
  description = "AKS total worker node 1 - 1000"
}

variable "worker_os" {
  default = "Ubuntu"
  description = "AKS worker os (Ubuntu/CBLMariner)"

}
variable "worker_maxpods" {
  default = 30
  description = "AKS maximum pods per node #Maximum pods per node is configurable up to 250"
}

variable "worker_encrypt" {
  default = false
  description = "AKS worker node encryption (true/false)"
}

variable "worker_type" {
  default = "VirtualMachineScaleSets"
  description = "AKS type of worker (AvailabilitySet/VirtualMachineScaleSets)"
}

variable "worker_disk_type" {
  default = "Managed"
  description = "AKS worker disk type (Ephemeral/Managed)"
}

variable "worker_size" {
  default = "Standard_D2_v2"
  description = "AKS vmware size for worker node"
}
```


kubernetes by Google

Feature on Project (Custom Config)

```
variable "worker_disksize" {
  default = "30"
  description = "AKS worker disk size (GB)"
}

variable "worker_sysctl_vmswappiness" {
  default = 0
  description = "AKS WORKER TUNNING for vm_swappiness (0 - 100)"
}

variable "worker_sysctl_vmvfscachepressure" {
  default = 50
  description = "AKS WORKER TUNNING for vm_vfs_cache_pressure (0 - 100)"
}
variable "worker_sysctl_fsfilemax" {
  default = 12000500
  description = "AKS worker tunning for fs_file_max (8192 - 12000500)"
}

variable "worker_sysctl_kernelthreadsmax" {
  default = 513785
  description = "AKS worker tunning for kernel_threads_max (20 - 513785)"
}

variable "worker_sysctl_netcorermemdefault" {
  default = 134217728
  description = "AKS worker tunning for net_core_rmem_default (212992 - 134217728)"
}

variable "worker_sysctl_netcorermemmax" {
  default = 134217728
  description = "AKS worker tunning for net_core_rmem_max (212992 - 134217728)"
}

variable "worker_sysctl_netcorewmemdefault" {
  default = 134217728
  description = "AKS worker tunning for net_core_wmem_default (212992 - 134217728)"
}
```

```
variable "worker_sysctl_netcorewmemmax" {
  default = 134217728
  description = "AKS worker tunning for net_core_wmem_max (212992 - 134217728)"
}

variable "worker_sysctl_netipv4tcpfintimeout" {
  default = 15
  description = "AKS worker tunning for net_ipv4_tcp_fin_timeout (5 - 120)"
}

variable "worker_sysctl_netipv4tcpkeepaliveprobes" {
  default = 9
  description = "AKS worker tunning for net_ipv4_tcp_keepalive_probes (1 - 15)"
}

variable "worker_sysctl_netipv4tcpkeepalivetime" {
  default = 7200
  description = "AKS worker tunning for net_ipv4_tcp_keepalive_time (30 - 432000)"
}

variable "worker_sysctlnetipv4tcptwreuse" {
  default = true
  description = "AKS worker tunning for net_ipv4_tcp_tw_reuse (boolean)"
}

#https://docs.microsoft.com/en-us/azure/aks/custom-node-configuration

variable "kubelet_logmaxline" {
  default = 200
  description = "AKS kubelet tunning for max container logs (> 2 line)"
}

variable "kubelet_logmaxmb" {
  default = 200
  description = "AKS kubelet tunning for max container logs before rotated (MB)"
}

variable "kubelet_imagegchighthreshold" {
  default = 60
  description = "AKS kubelet high threshold for start GC image (0% - 100%)"
}
```


kubernetes by Google

Feature on Project (Custom Config)

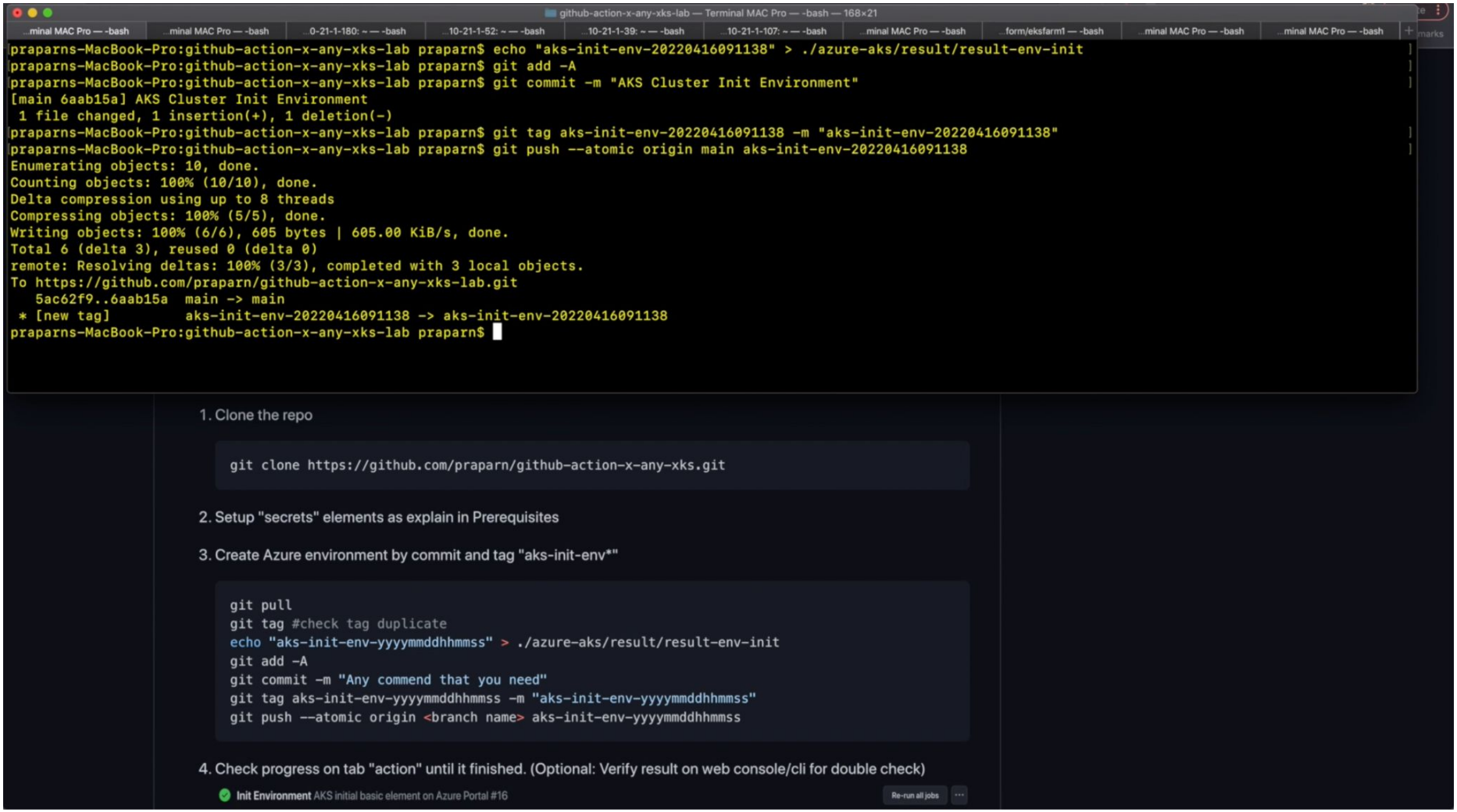
```
variable "kubelet_imagegclowthreshold" {
  default = 10
  description = "AKS kubelet low threshold for start GC image (0% - 100%)"
}

variable "network_policy" {
  default = "calico"
  description = "AKS network policy for use in cluster. (calico/azure)"
}

variable "network_plugin" {
  default = "kubenet"
  description = "AKS network plugin for cluster. (kubenet/azure)"
}

#https://docs.microsoft.com/en-us/azure/aks/custom-node-configuration

variable "load_balancer_sku" {
  default = "standard"
  description = "AKS SKU for load balancer. (basic/standard)"
}

variable "load_balancer_profile" {
  default = "standard"
  description = "AKS SKU for load balancer profile. (basic/standard)"
}

variable log_analytics_workspace_name {
  default = "template-log-aks-lab-2022"
  description = "Log workspace name. This will be add suffix with random number automatically"
}

variable log_analytics_workspace_sku {
  default = "PerGB2018"
  description = "SKU for pricing of log analytics on Azure (https://azure.microsoft.com/en-us/pricing/details/monitor/)"
}

#  AKA Properties #
```

Demo Session

K8S Lifecycle with Github Action

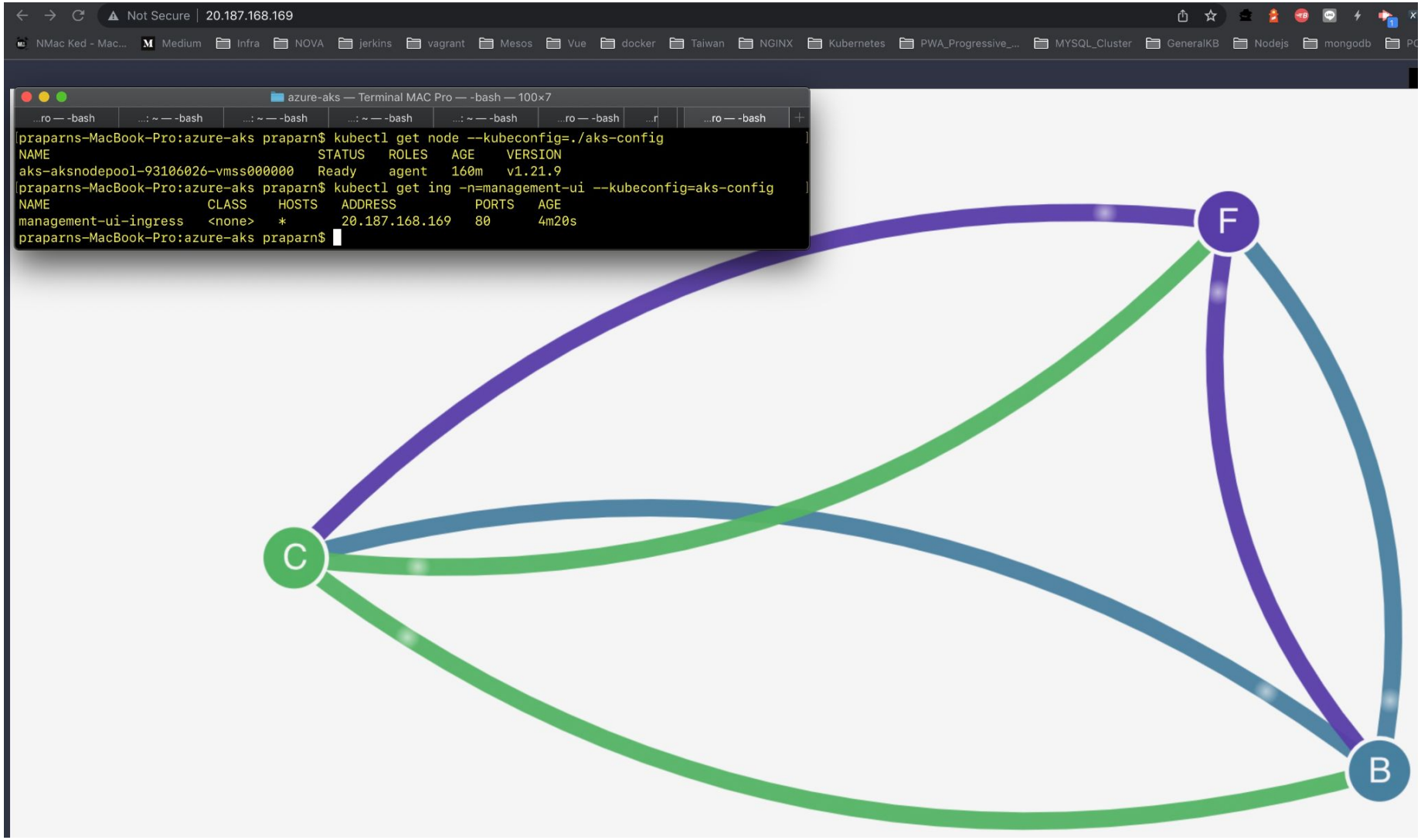
kubernetes by Google

Init Environment



kubernetes by Google

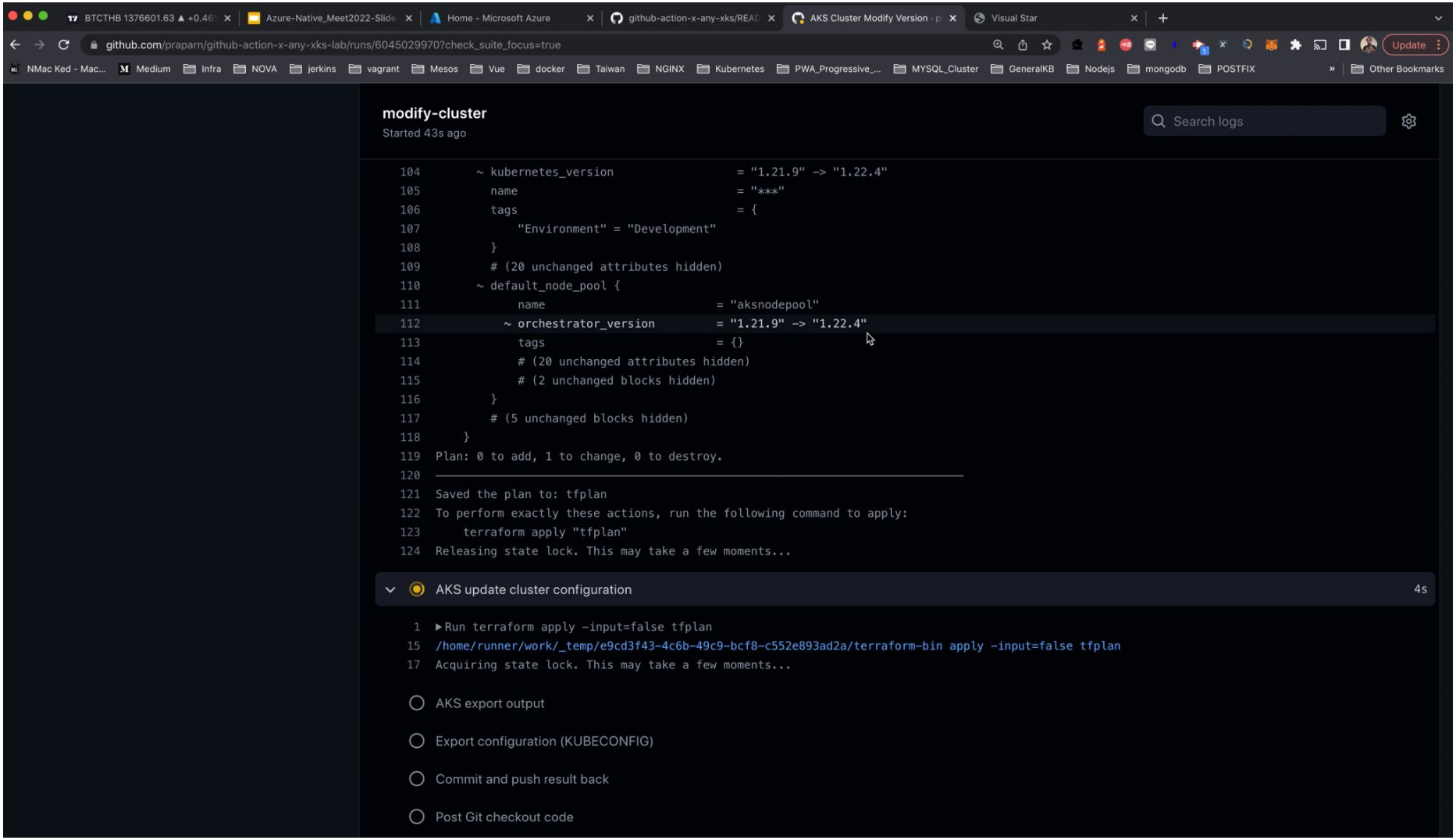
Create Cluster



K8S Lifecycle with Github Action

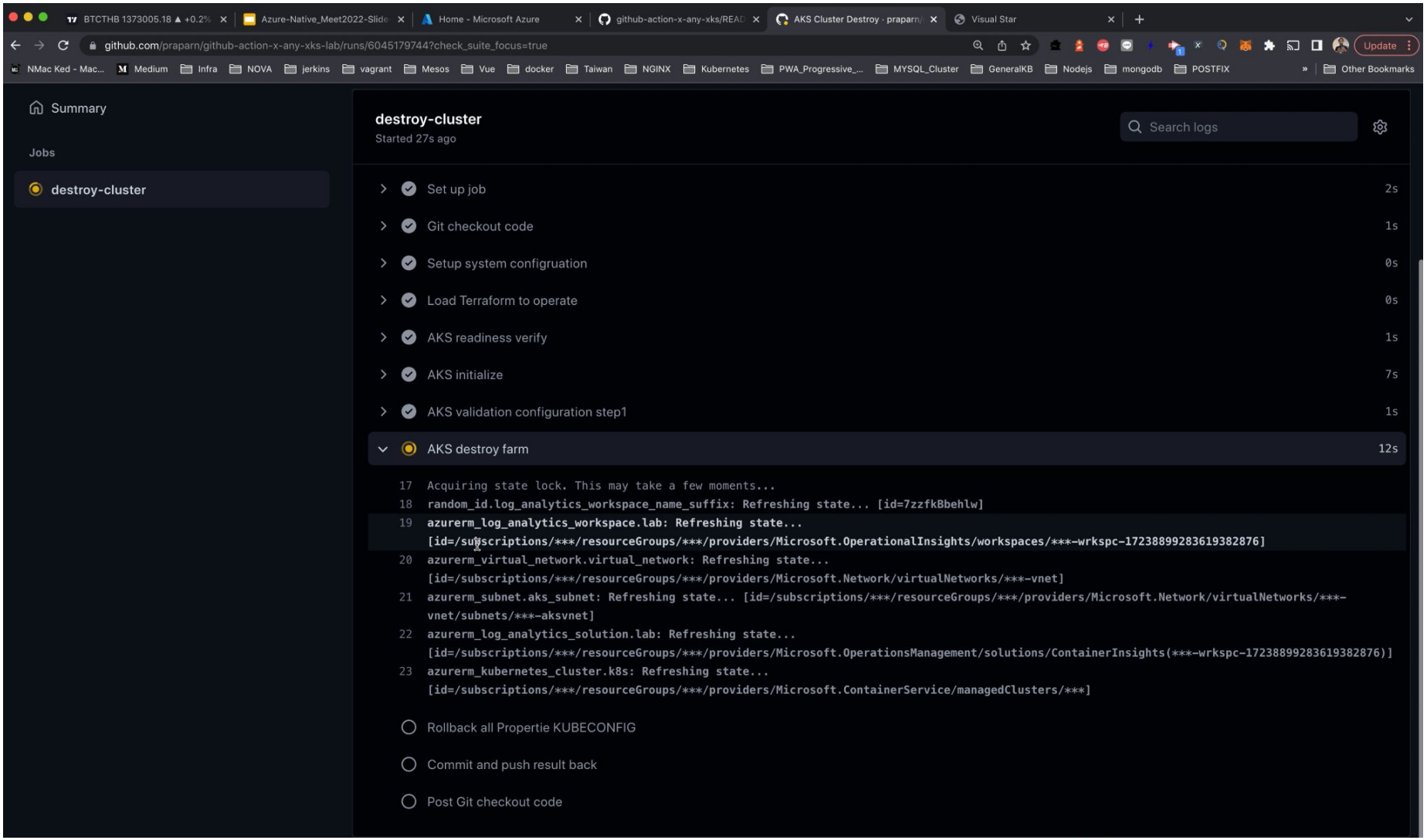
kubernetes by Google

Modify Cluster



kubernetes by Google

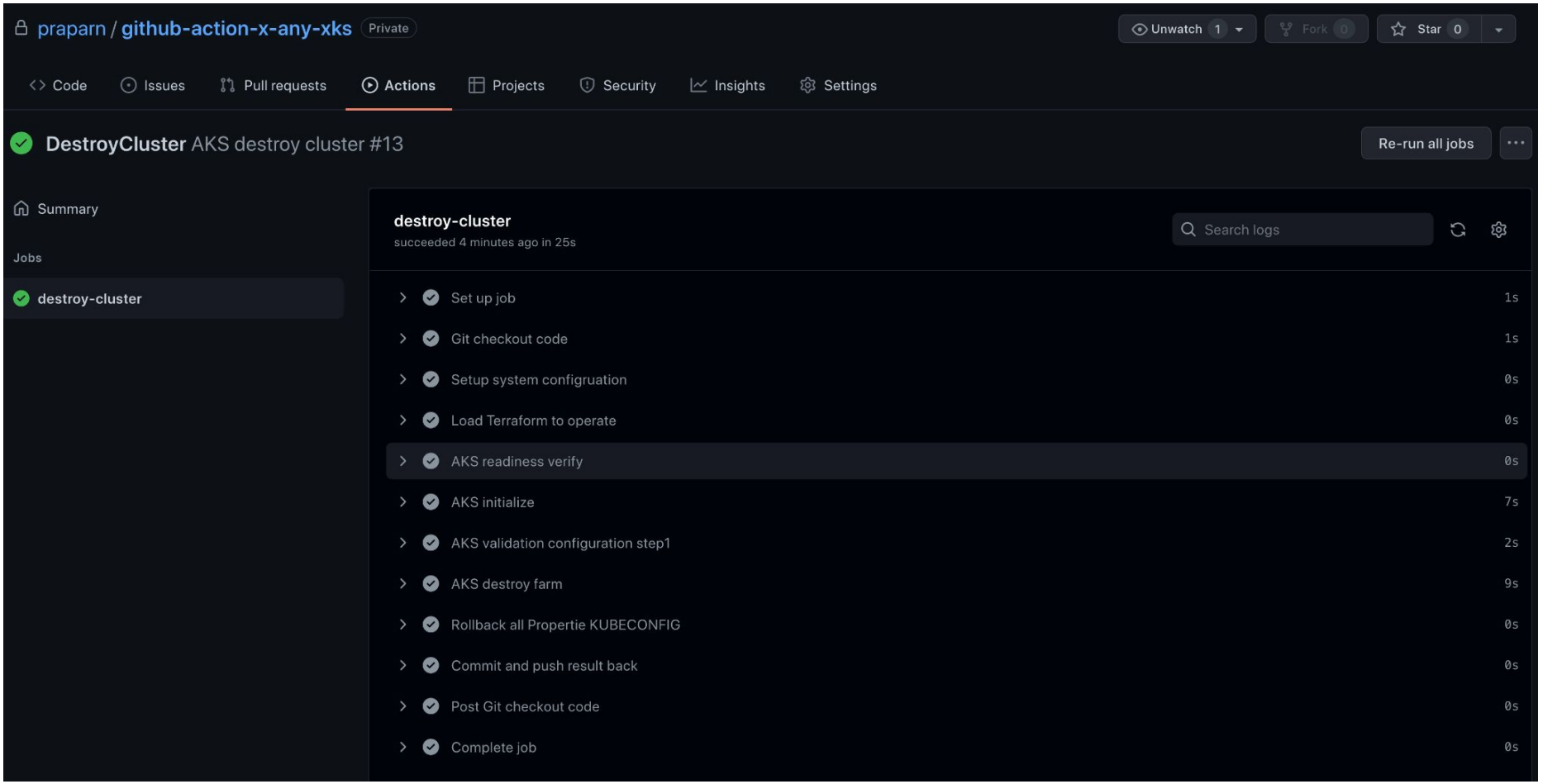
Destroy Cluster



K8S Lifecycle with Github Action

kubernetes by Google

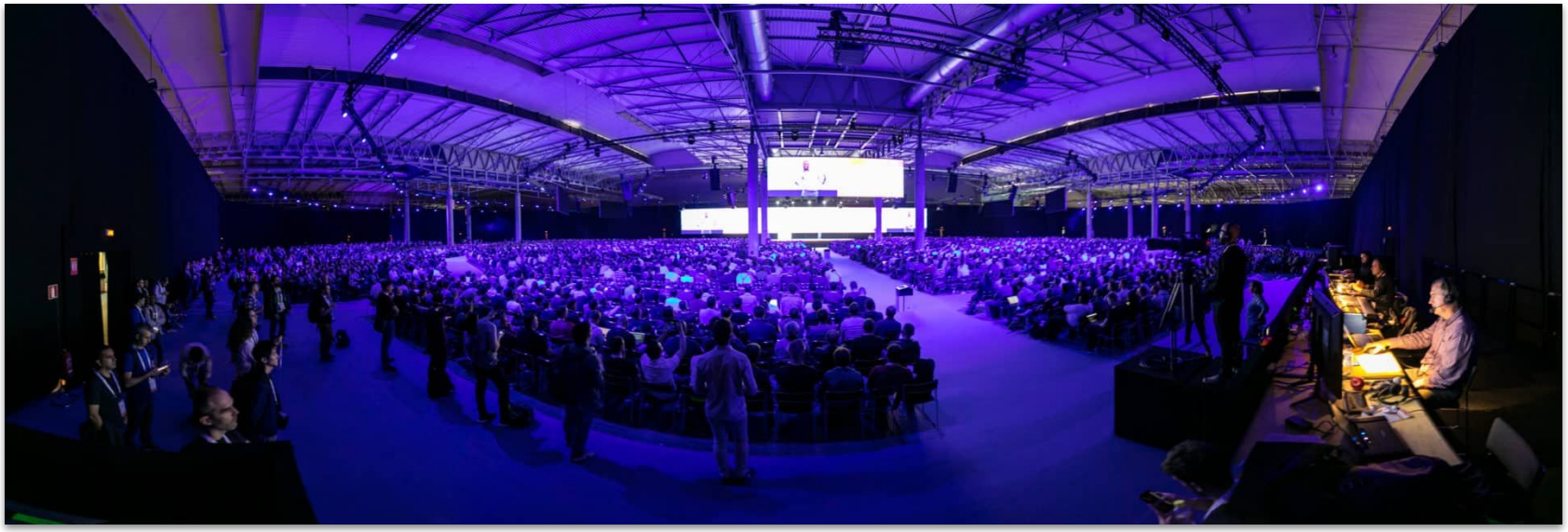
Destroy Environment

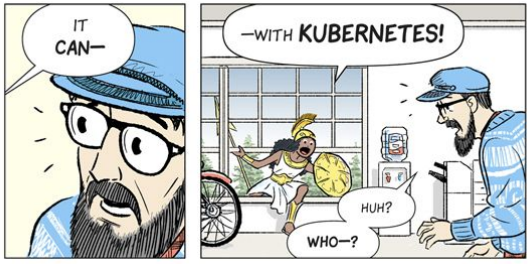


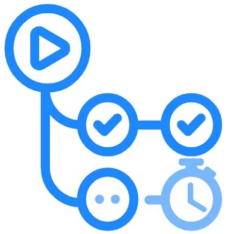
kubernetes by Google

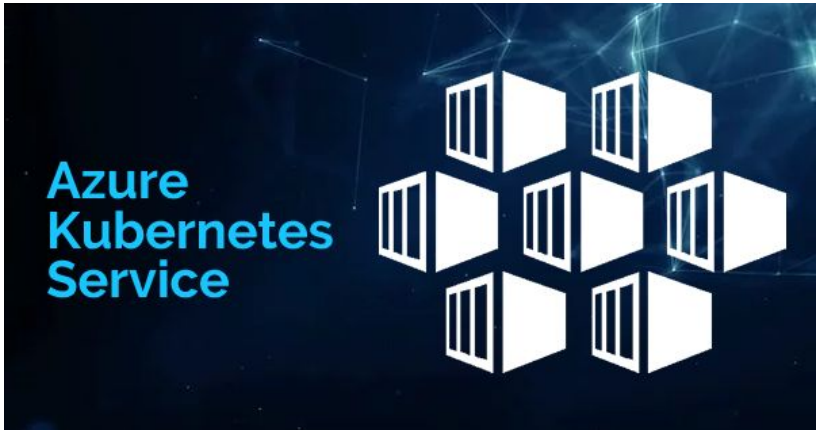
Q&A

kubernetes by Google









Thanks You

kubernetes by Google