KubeCon | CloudNativeCon

North America 2024

Yahoo's Kubernetes Journey On-Prem to Multi-Cloud

Nandhakumar Venkatachalam

Payal Patel

Agenda

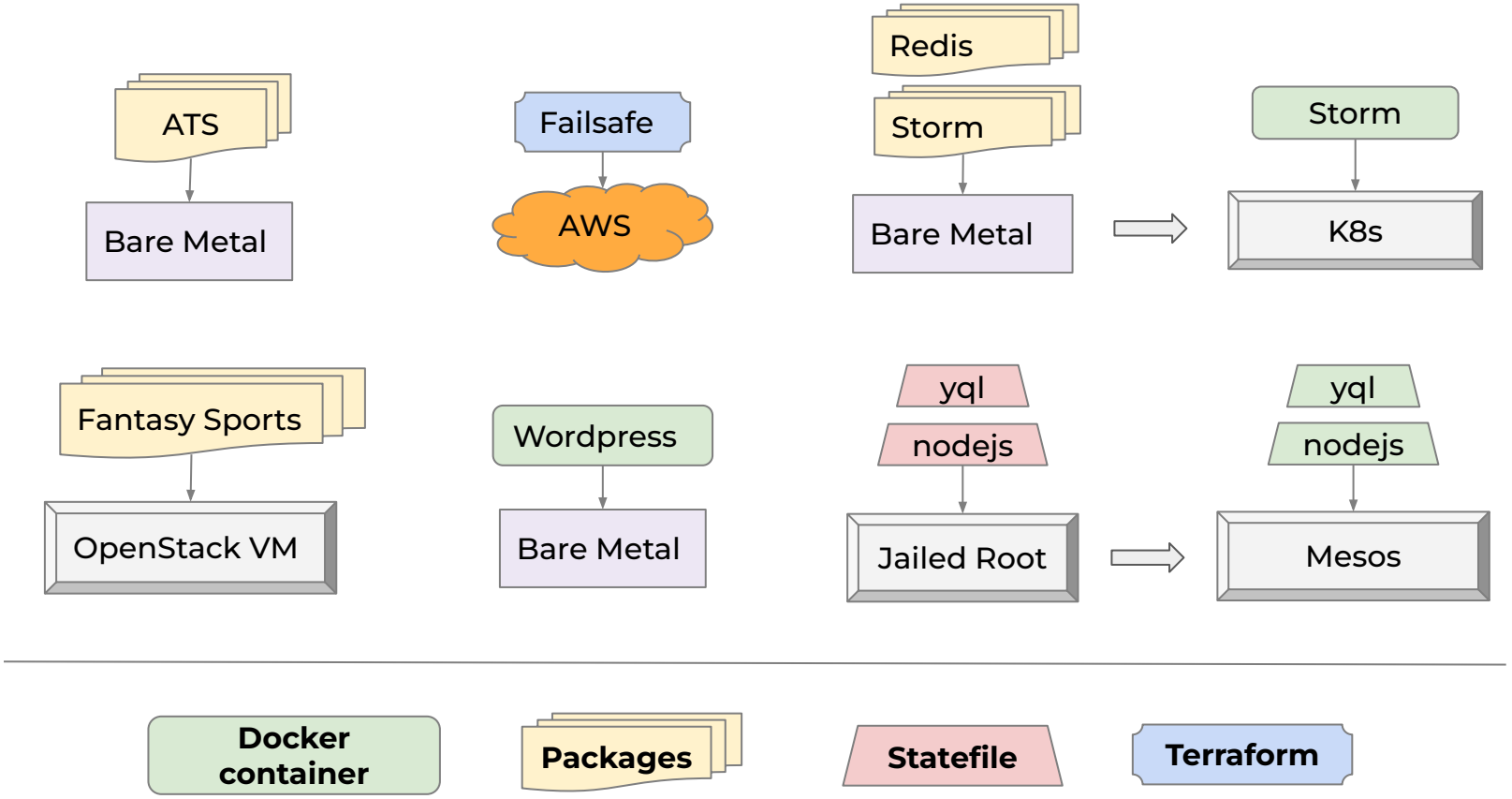
- The Platform
- User Interface
- Onprem Adventure
- Digital Transformation
- Cluster Management
- App Security
- Key Takeaways

KubeCon | CloudNativeCon

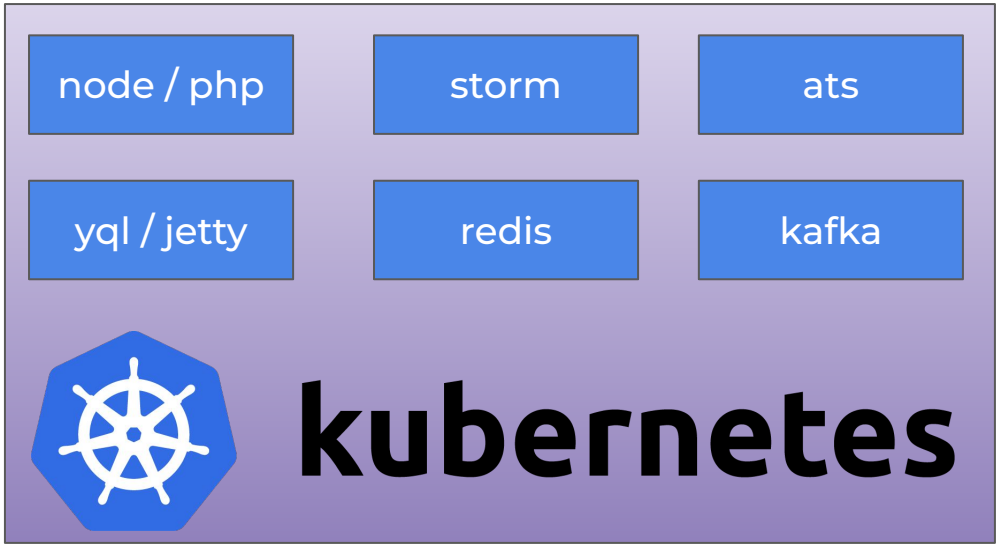
North America 2024

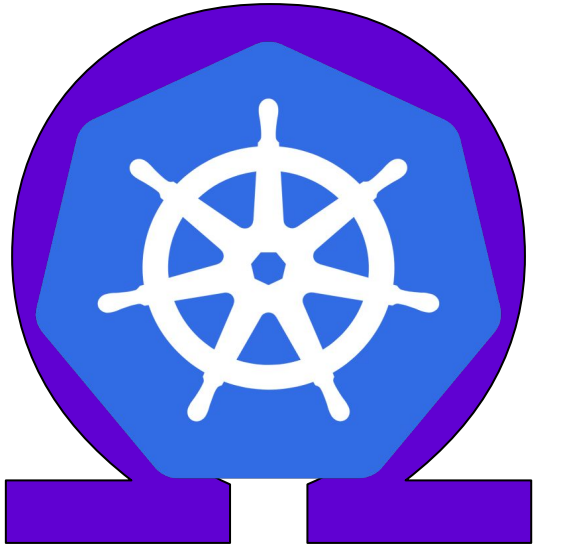
The Platform

Tech landscape @ 2016



Platform Team





Omega

Omega Platform

- One team focusing on everything Kubernetes
- Solve common problems affecting many teams
- Lightweight set of tools and templates that make it straightforward to deploy Kubernetes workloads
- Take care of common use cases such as Identity, Routing, Security, Observability
- Handles per-environment (prod/nonprod) and per-data–center (region) differences, namespace & access control
- Full Kubernetes interface is always available to users

Omega

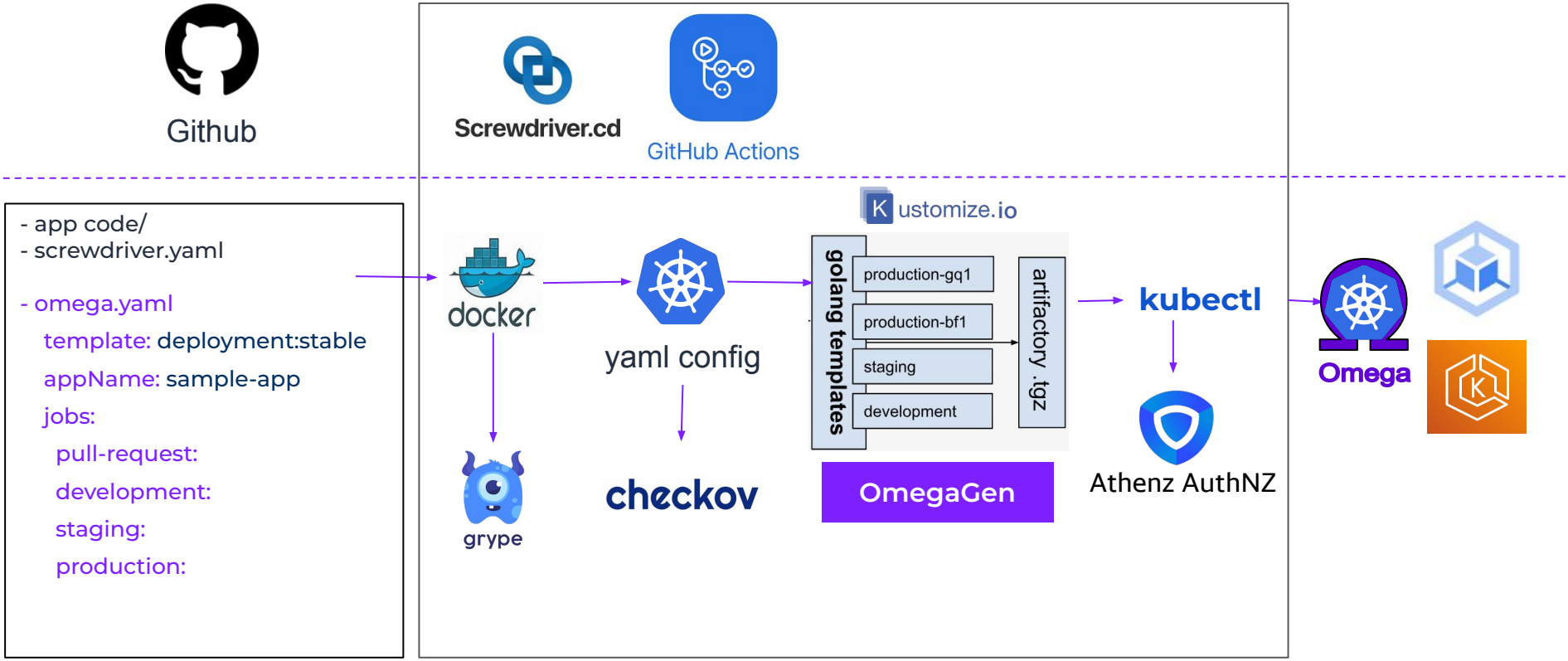
KubeCon | CloudNativeCon

North America 2024

User Interface

.yaml

Developer Focus vs Platform focus



Without Standard Tool


```
git_repo="awesomeapp"
cd ${git_repo}

# APP
awesomeapp/main.go

# CI/CD pipeline definition
screwdriver.yaml

# K8S deployment definition
deploy_target/omega/omega.yaml

# optional scripts
deploy_target/omega/scripts/
  - app_configure #build image
  - app_start     #app start
```

```
- autoscale.yaml
- deployment.yaml
- podDisruptionBudget.yaml
- service.yaml
- serviceaccount.yaml
- istio/
    - destinationRule.yaml
    - gateway.yaml
    - peerAuthentication.yaml
    - requestAuthentication.yaml
    - sidecar.yaml
    - virtualservice.yaml
- values/
    - qa-gq1.yaml
    - prod-bf1.yaml
    - prod-gq1.yaml
- Dockerfile
```

~20 YAML files

```
- steps:

    - publish:
        go build
        docker build
        helm template
        tar -cvf [k8s-yaml.tar]
        scp k8s-yaml.ta [registry]
        meta set [app version]
    - deploy:
        meta get [app version]
        curl -O .zip
        unzip
        kubectl-ctx
        kubectl apply -f
        kubectl rollout status
        curl https://.../status.html
        functional-test.sh
```

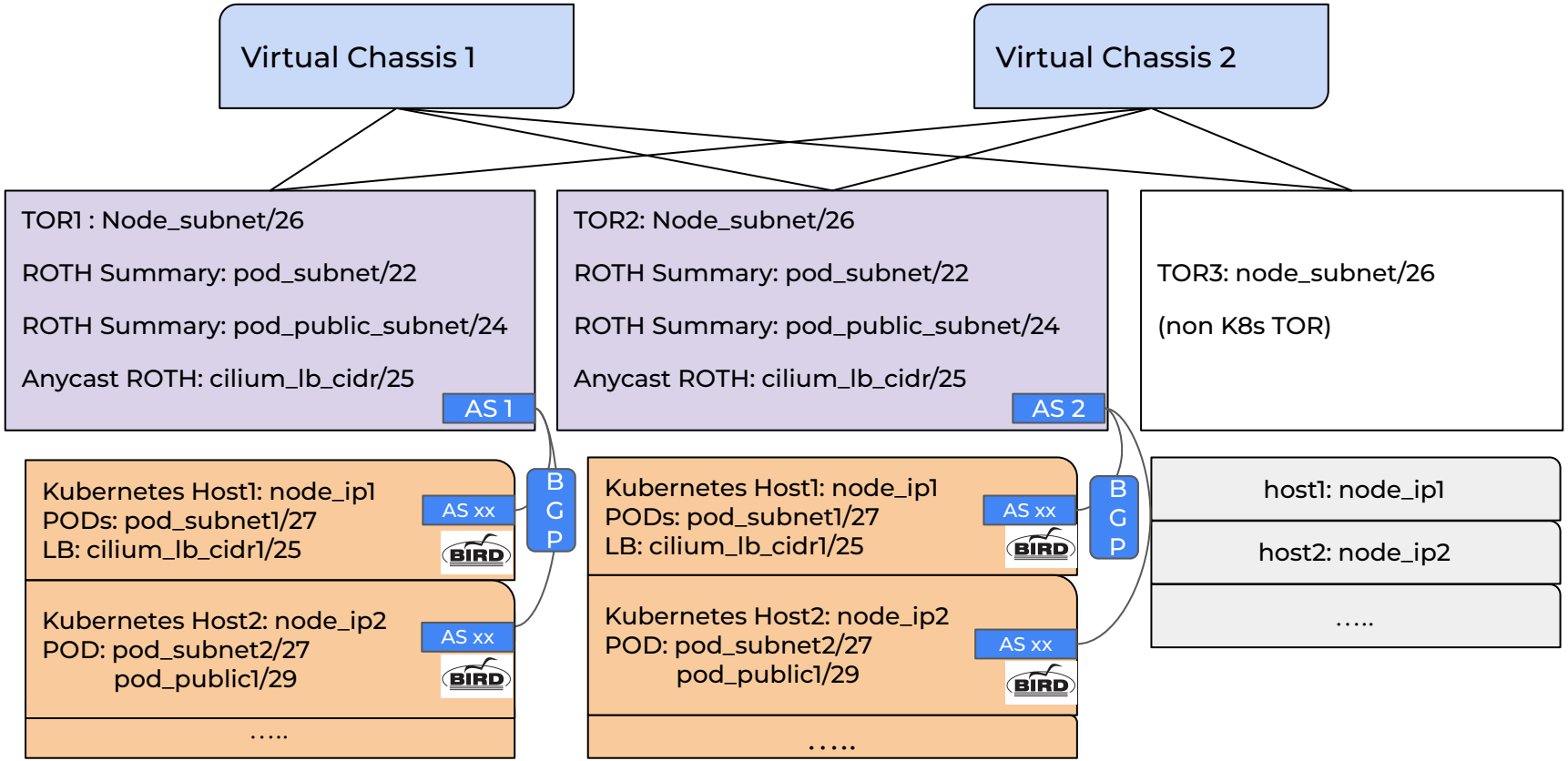
Multiple Steps

KubeCon | CloudNativeCon

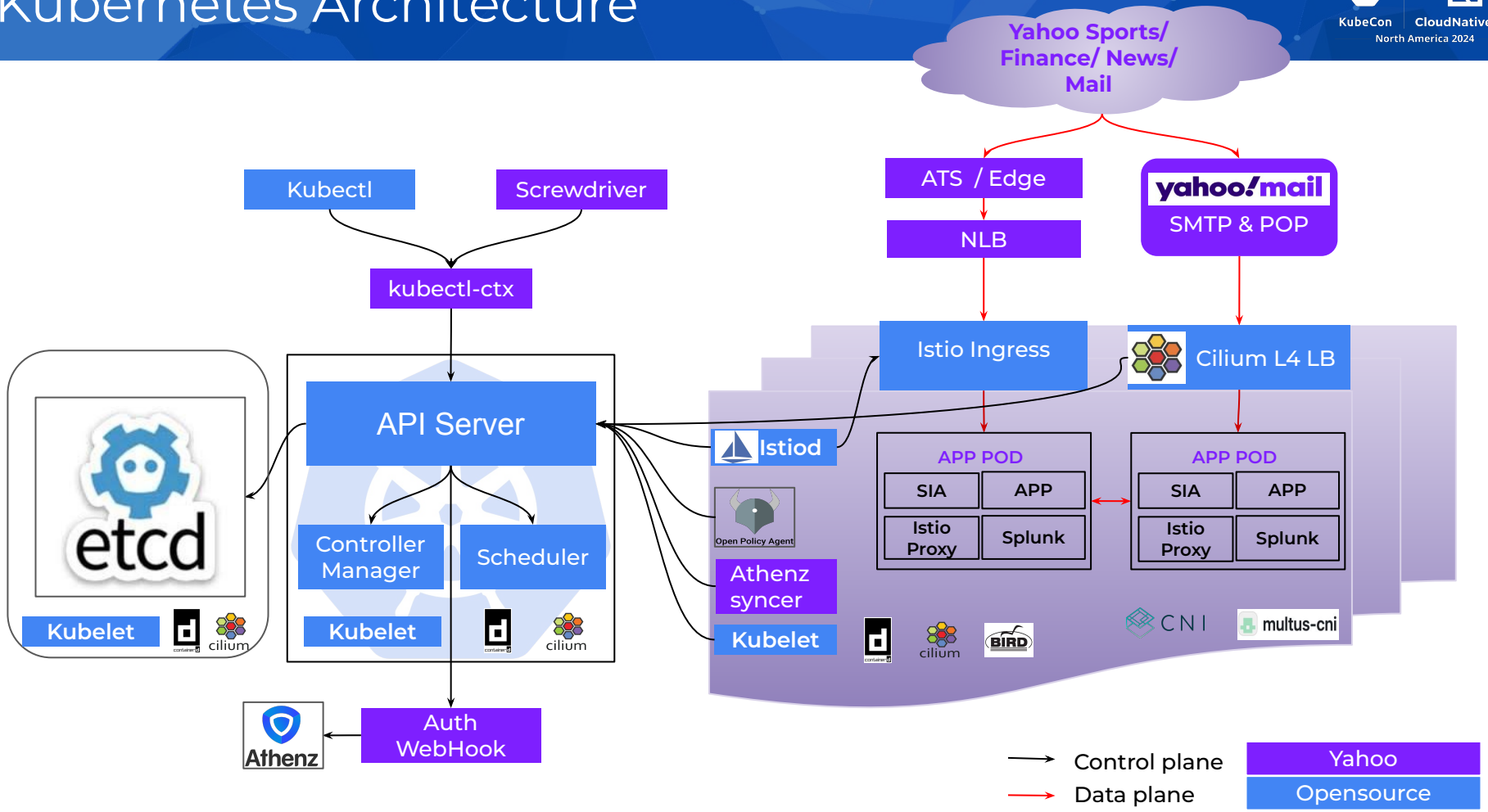
North America 2024

Onprem Adventure

Kubernetes Networking



Kubernetes Architecture



Stacks and Numbers

O11Y splunk>

Operator APACHE STORM redis STRIMZI

Storage Dell PowerFlex TRIDENT

Network cilium Istio CNI multus-cni

Kubernetes Athenz Open Policy Agent etcd

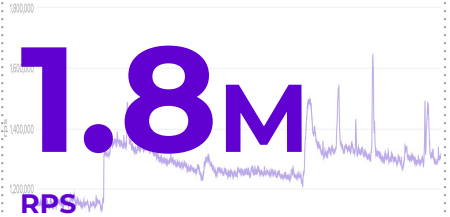
Baremetal ANSIBLE BIRD containerd kata containers NVIDIA

37
Clusters

8K
Nodes
2 clusters each w/
1.2K nodes 68K CPU Cores

1.5K
omega.yaml apps

135K
Pods
replicaset: 50%
statefulset: 10%
daemonset: 40%



385K
Containers

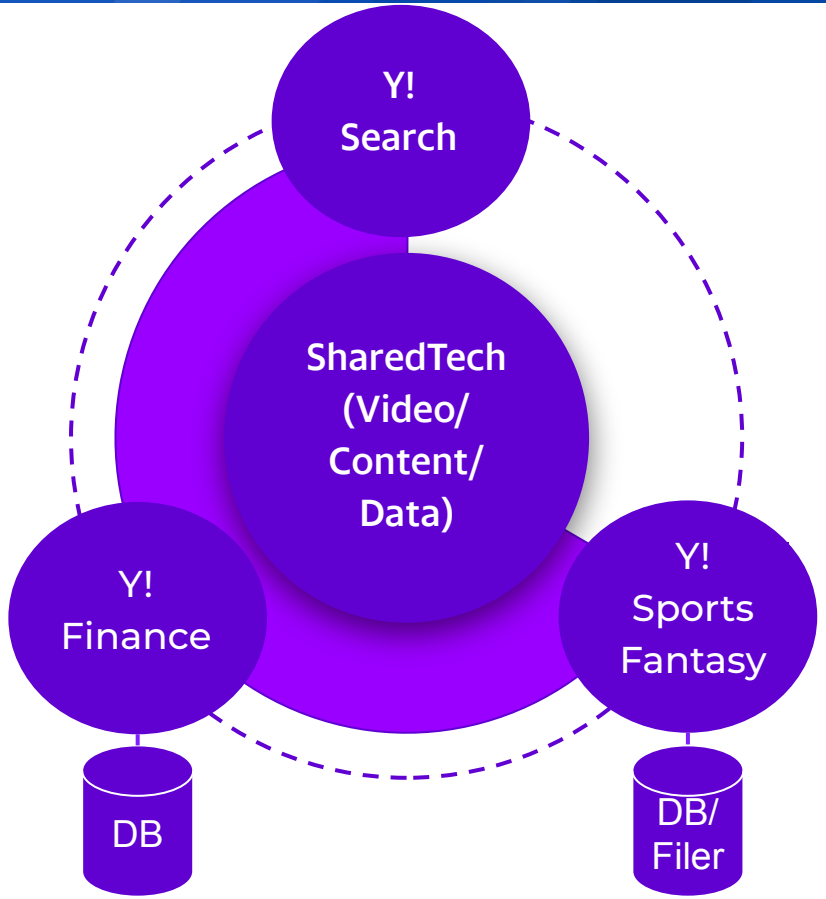
KubeCon | CloudNativeCon

North America 2024

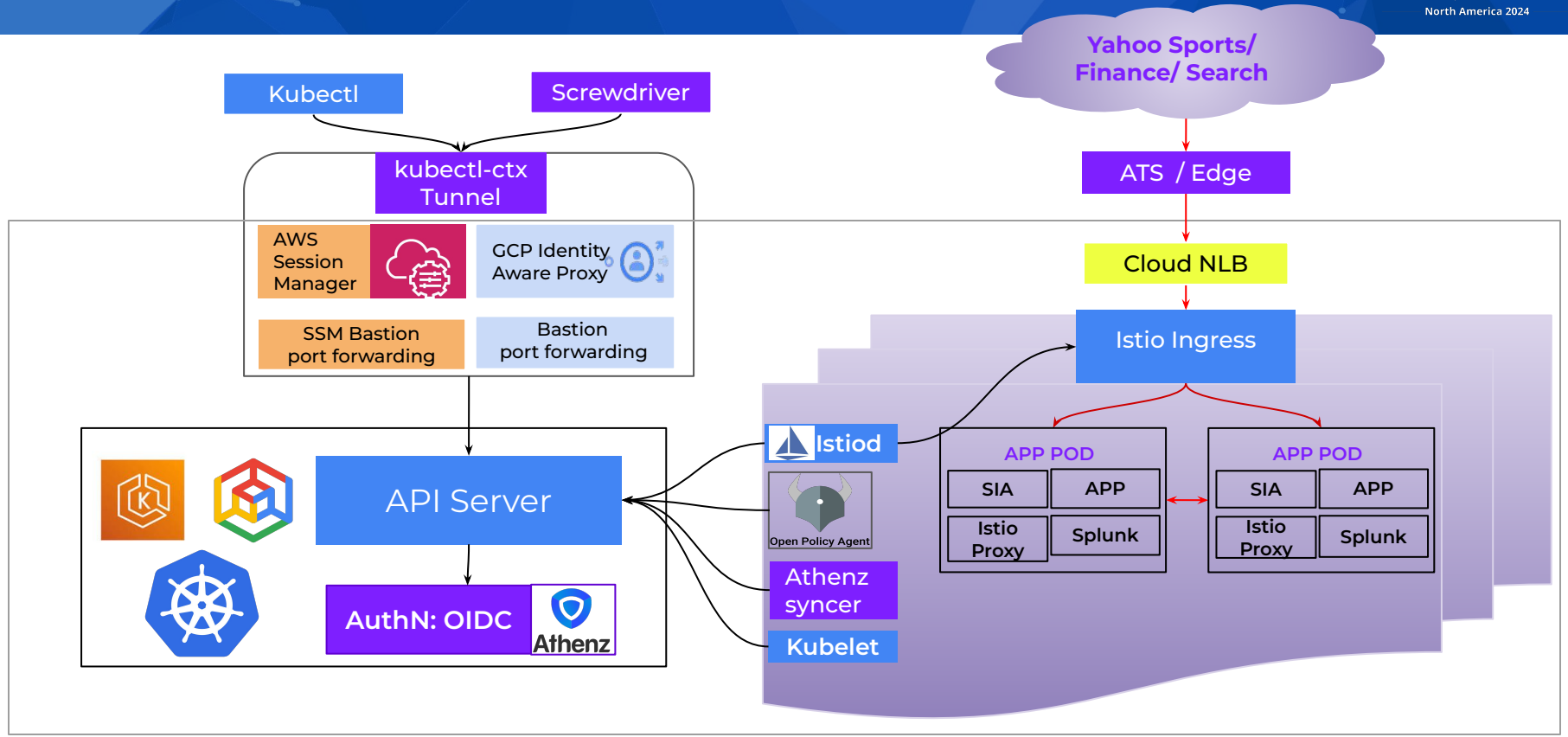
Digital Transformation 2023

Approach to Cloud

- Separate Account and VPC per product unit
- Provision Kubernetes per region
- Provides better cost allocation and maintenance
- Managed Control plane
- Secure access to Kubernetes API
- Networking Infrastructure



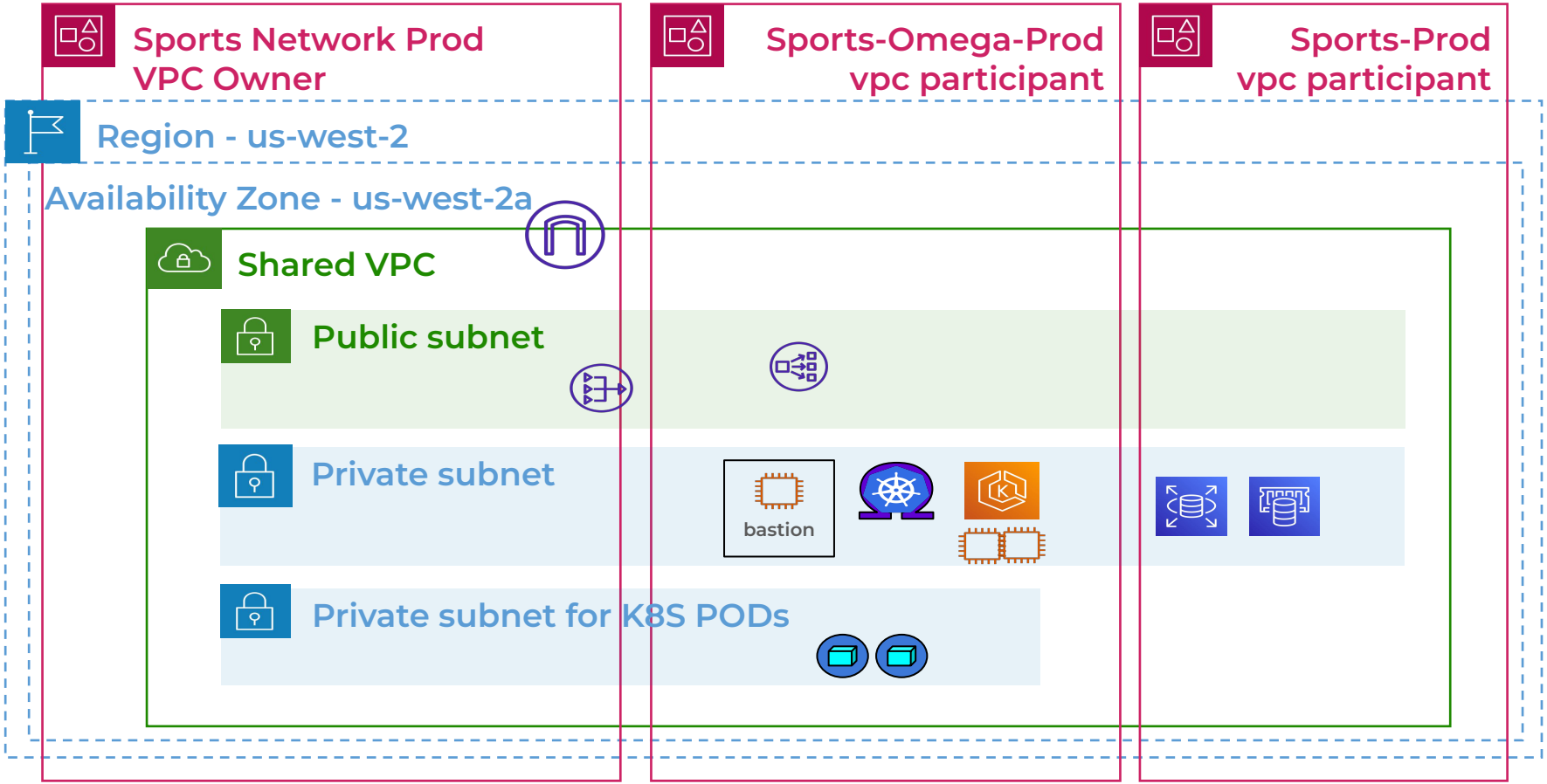
Cloud Cluster Architecture - GKE + EKS



VPC Networking

- Manage centralized network infrastructure in cloud
  - Standardize network features
  - Use same AZs across all VPC, minimize cross-AZ charges
  - Common Modules for VPC Peering, Endpoint, Firewall
- Sufficient IP Capacity
  - Every VPC has deconflicted IP ranges
- Efficient K8S networking
- Prod and Non-prod separation

AWS Shared VPC Architecture



EKS K8S Networking

- Leverage separate CIDR range for POD IPs (ex: rfc6598 100.64.0.0/10)
- Enable HIGH IP Density for Pods, fewer NMU and ec2 metadata call
  - AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG=true
  - Create VPC CNI ENIConfig and reference POD Subnet
  - ENABLE_PREFIX_DELEGATION=true
    - Trade off SGP (security group for pods), K8S network policy is an alternate
  - Each node is assigned minimum one /28 prefix for pods, max of seven /28's to get 110 pods per node

Subnet size per AZ	IPs Available	7	6	5	4	3	2	1	Prefix List Count
/18	16384	146	170	204	256	341	512	1024	Node Count

- Use fixed service IP range like a /16

GCP Shared VPC Architecture

Host Project
net-prod

Shared VPC

region: = us-west1

primary cidr: x.x.x.x/19

pod_cidr: x.x.x.x/18

service_cidr: 172.20.0.0/16

Cloud Firewall Rules

Service Project 1
omega-prod

Bastion

Load balancer

GKE

Nodes

Service Project 2
app-service-prod

Memorystore

GKE K8S Networking

- By default GKE assigns /24 alias (256 IPs, 2 x max_pods) from secondary CIDR
  - Consider max_pods to control alias IP assignment
- Service CIDR caveats
  - Tracked as part of subnet's secondary range
  - Remove from secondary range after cluster creation if using fixed CIDR range
- Plan for multiple PSA and PSC subnets of smaller range
- GCP Subnets are region-specific unlike AZ-specific in AWS

Primary	**/19**	**for Nodes and other shared vpc services; 8188 usable IPs**
Secondary - Pod subnet per region	**/18**	**64 nodes with /24 per node (max pod per node 110)** **128 nodes with /25 per node (max pod per node 64)**
K8s Service	**/16**	**Consider using fixed CIDR for every cluster**

KubeCon | CloudNativeCon

North America 2024

K8s Cluster Management

Tools to manage clusters in Cloud

- Need for tools to manage and create K8s clusters in cloud
- FLEKS - internally developed tool to manage EKS clusters
  - Built using Ruby
  - Uses CloudFormation to manage cloud resources in AWS
  - Only supports AWS EKS
- KDNC - built initially for GKE use case in 2023
  - Built using Go
  - Uses Terraform to manage cloud resources
  - Supports both GKE and EKS clusters

Common requirements for setting K8s cluster

- Prerequisite - VPC / network setup
- Simple configuration interface
  - Provide a cluster definition in a single unit/file
- Easy deployment
  - Build and manage cluster using CI/CD
- Customize cluster using addons
  - Configure cluster-related add-ons/plugins in a single interface

2019 - EKS Cluster Setup using FLEKS


```yaml
cluster:
  name: finance
  stage: prod1
  version: "1.29"

global-tags:
  corp:Environment: prod

athenz:
  domain: finance.aws.prod.omega

aws:
  account-id: "xxxxx"
  region: us-east-1
  vpc-id: vpc-0123456789

  subnets:
    public:
      - subnet-01234567890123456
      - subnet-98765432109876543
    private:
      - subnet-00112233445566778
      - subnet-99887766554433221

core-addons:
  vpc-cni:
    version: "v1.16.4-eksbuild.2"
    configuration:
      env:
        AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG: 'true'
        ENABLE_PREFIX_DELEGATION: 'true'
        ENI_CONFIG_LABEL_DEF: 'topology.kubernetes.io/zone'
  coredns: {}
  aws-ebs-csi-driver: {}

addons:
  aws-load-balancer-controller:
    ref: v4.1.2
  karpenter:
    params:
      fargate-profile: fargate
  cloudability-metrics-agent:
  external-dns:
  metrics-server:
  node-local-dns:
  athenz-oidc-rbac:
  aws-efs-csi-driver:
```

2023 - GKE Cluster Setup using KDNC


```
project_id                      = "omega-myproject-s"
network                         = "myproject-nonprod1-vpc"
network_project_id              = "net-myproject-s"
subnetwork                      = "myproject-nonprod1-subnet-1"
ip_range_pods                   = "pod-subnet"
ip_range_services               = "service-subnet"
region                          = "us-east1"
zones                           = ["use1b", "use1c"]
cluster_name                    = "myproject-np1-use1"
enable_identity_service         = true
athenz_domain                   = "myproject.gcp.nonprod.omega"
kubernetes_version              = "1.29.6-gke.1038001"
node_pools_labels = {
  general = {
    dedicated = "general"
  }
}
node_pool = [
  {
    name              = "general"
    min_count         = 2
    max_count         = 100
    spot              = false
    machine_type      = "n2-standard-32"
    max_pods_per_node = 32
    disk_size_gb      = 500
    node_version      = "1.29.6-gke.1038001"
  }
]
node_pools_oauth_scopes = {
  general  = ["https://www.googleapis.com/auth/cloud-platform"]
}
monitoring_enabled_components = ["SYSTEM_COMPONENTS", "APISERVER", "SCHEDULER", "CONTROLLER_MANAGER"]
logging_enabled_components    = ["SYSTEM_COMPONENTS", "APISERVER", "CONTROLLER_MANAGER", "SCHEDULER", "WORKLOADS"]
```

2024 - EKS Cluster Setup using KDNC


```
1  cluster_name       = "omega-npuse1"
2  vpc_id             = "vpc-xxxxxxxxxxxxxxxxx"
3  subnet_ids         = ["subnet-1xxxxxxxxxxxxxxxx", "subnet-2xxxxxxxxxxxxxxxx", "s
4  region             = "us-east-1"
5  aws_account_id     = "123456789012"
6  athenz_domain      = "k8s.omega.aws-nonprod"
7  bastion_name       = "bastion-use1"
8  ami_release        = "AL2023"
9  architecture       = "x86_64"
10 ami_type           = "STANDARD"
11 kubernetes_version = "1.29"
12
13 # Self Managed Node Group(s)
14 self_managed_node_groups = [
15   {
16     name          = "general"
17     desired_size  = 1
18     max_size      = 2
19     min_size      = 1
20     instance_type = "c6i.2xlarge"
21   }
22 ]
23
24 vpc_filters = [
25   {
26     name   = "vpc-id"
27     values = ["vpc-xxxxxxxxxxxxxxxxx"]
28   }
```

KubeCon | CloudNativeCon

North America 2024

Cluster Maintenance

Cluster Upgrades - onprem

- On-prem K8s version upgrade
  - Review changelog summary
  - Use node manager k8s operator to automatically do upgrades in a controlled manner ensuring capacity is maintained without impacting live traffic in production
  - Can take few months due to the scale on-prem (~8000+ physical hosts)
  - Same physical hosts has to be upgraded
- Combined upgrade of K8s version and node OS

Cluster Upgrades - cloud

- K8s version upgrade on Cloud (EKS + GKE)
  - Ensure the specific version is being used instead of using auto upgrades though GKE will automatically upgrade if that targeted version is being eoled.
  - Need to ensure the istio matches the version with K8s version
  - Easy upgrade due to elastic nature of the hosts in cloud and auto scale up accordingly unlike onprem
  - Can handle node capacity automatically before/during the upgrade

Cluster Monitoring - onprem

- On-prem cluster monitoring
  - Prometheus to store metrics for the cluster and workloads
  - Alertmanager for alerting on a significant events to ensure stability of the cluster
  - Grafana for visualization of the metrics and data over time
  - Prometheus federation for multi-cluster visualization of the metrics across 7 data center
  - Splunk for logging and alert based on certain events

Cluster Monitoring - cloud

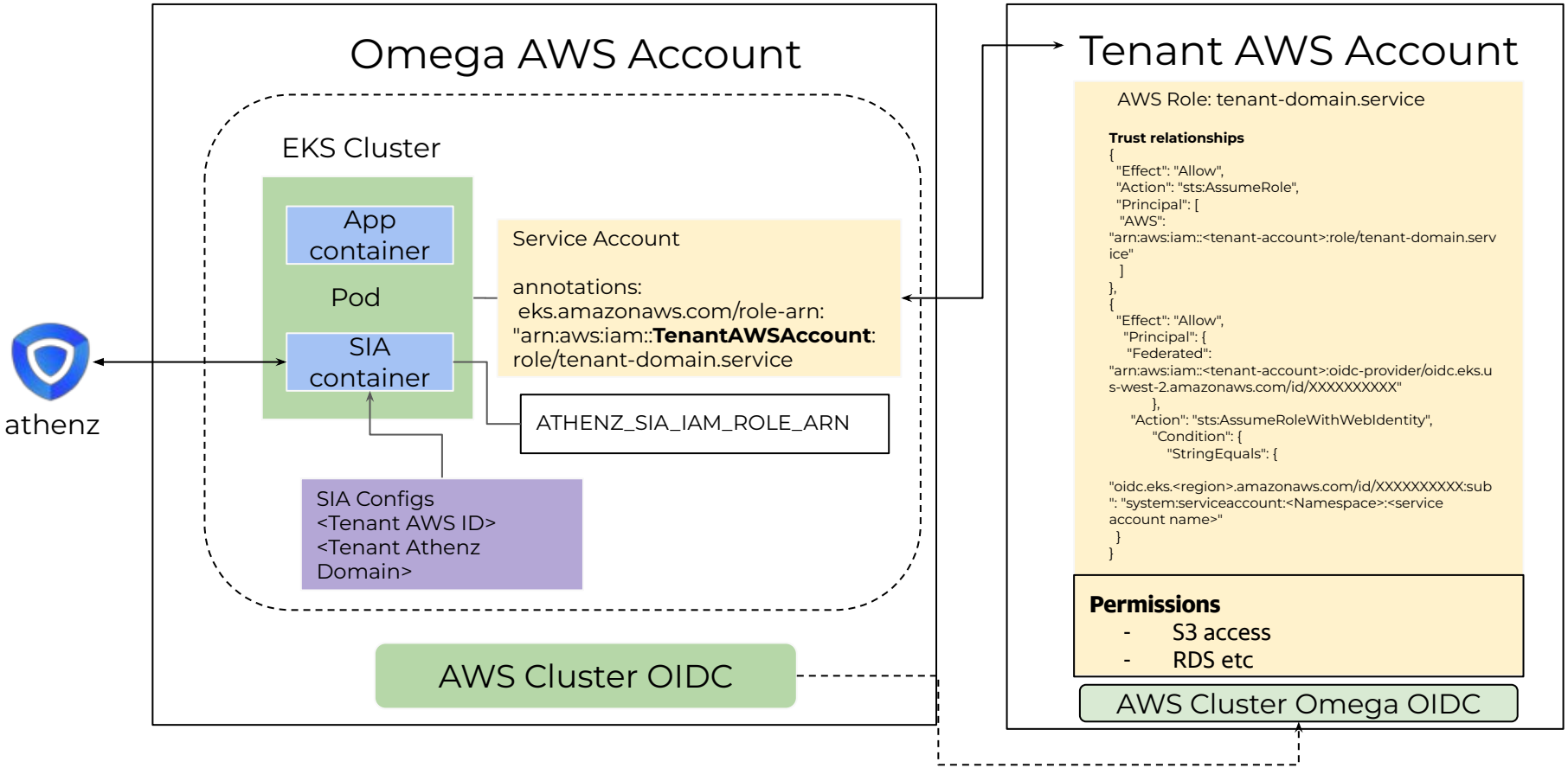
- Cloud cluster monitoring
  - Used the same components as on-prem, leveraged existing components seamlessly using Prometheus, Alertmanager & Grafana
  - Leveraged out of the box GKE Observability solution for tracking events for application workload

KubeCon | CloudNativeCon

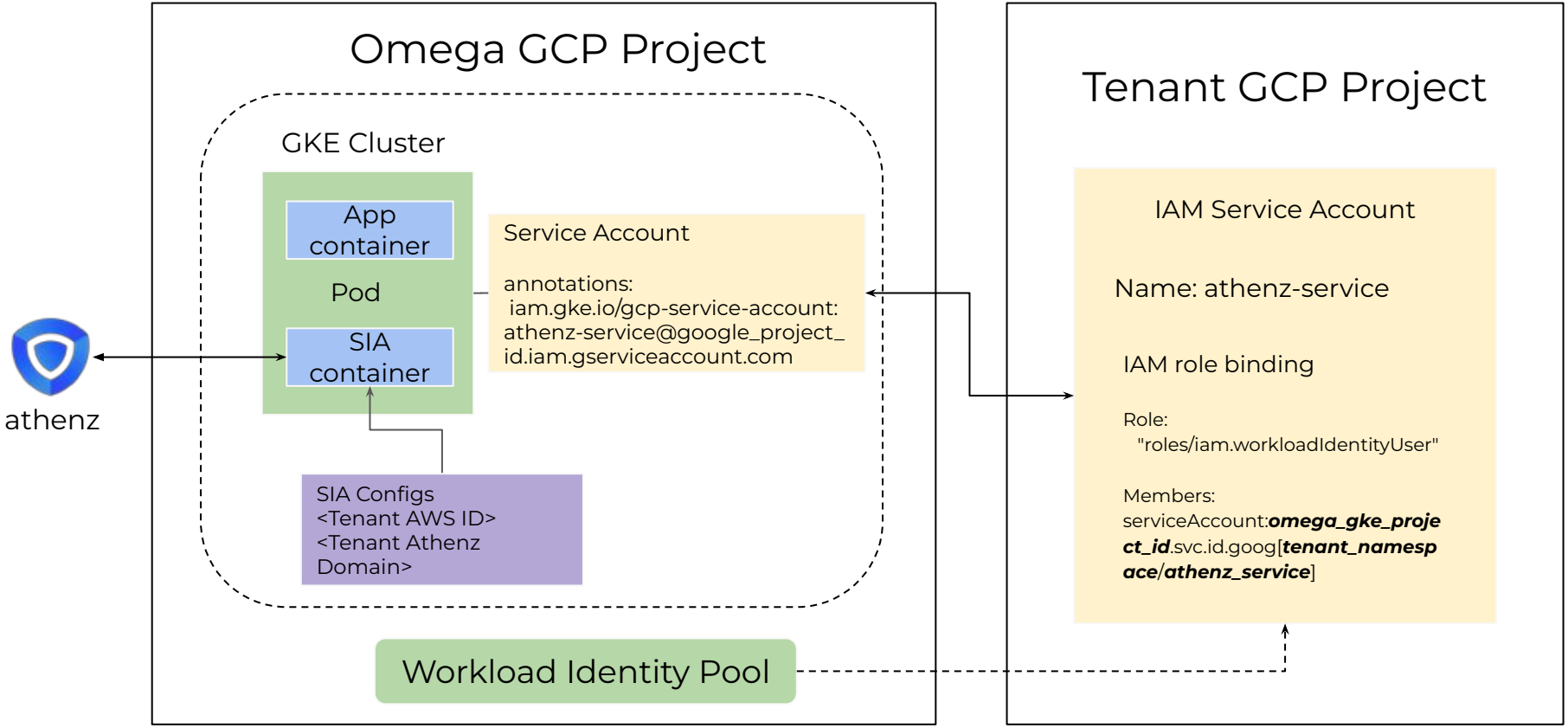
North America 2024

App Security

App / service Identity on EKS



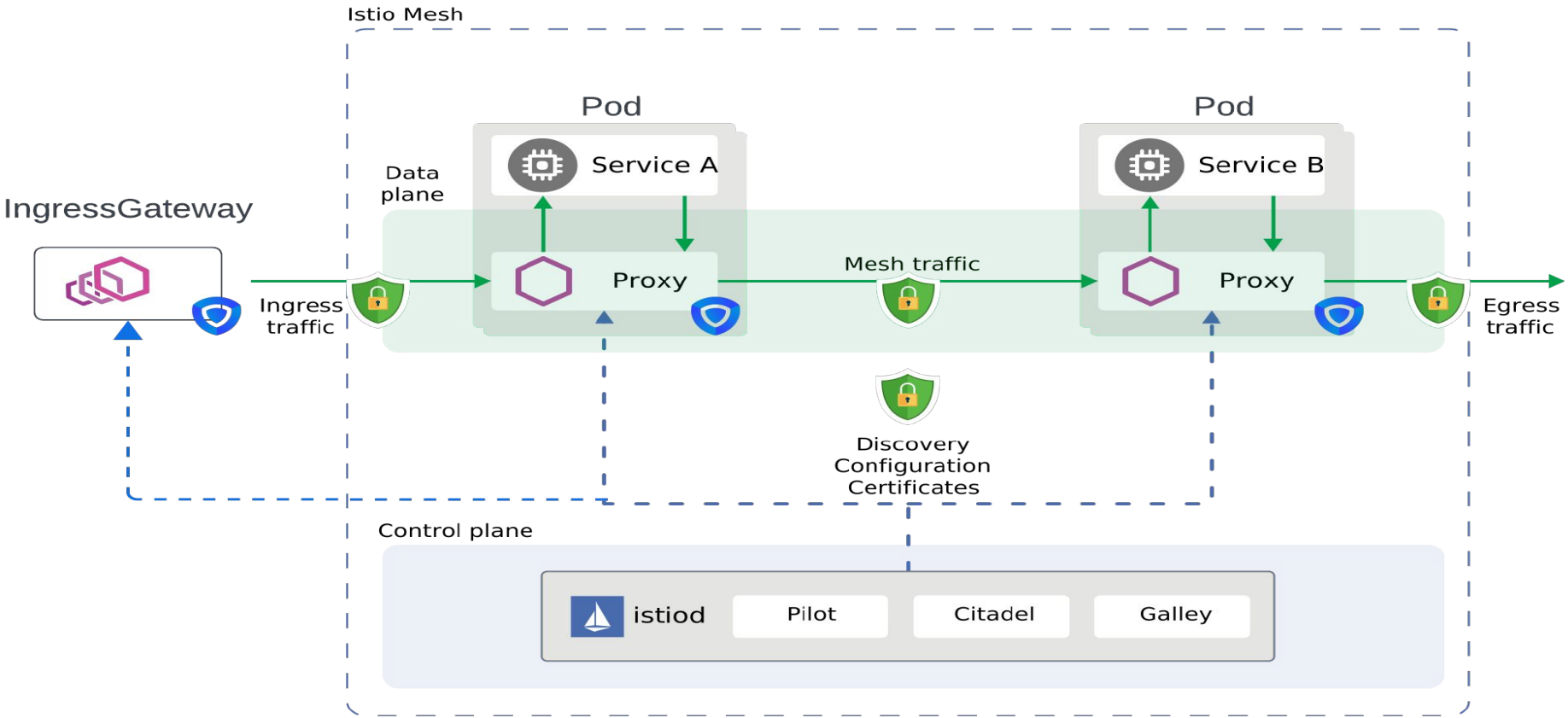
App / service Identity on GKE



App Identity terraform module on Omega K8s on Cloud

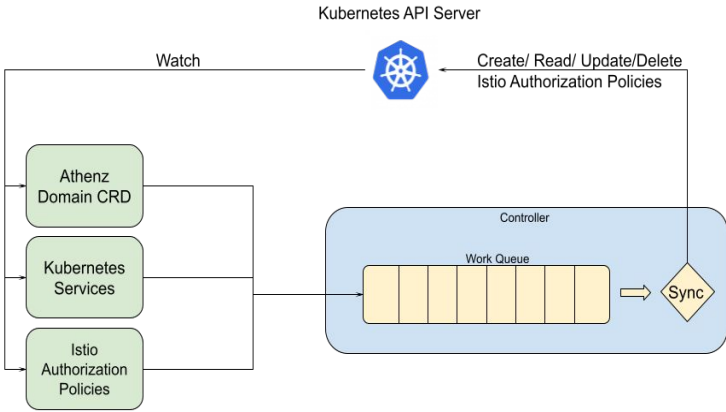

```
# EKS identity module
module "tf_omega_eks_identity" {
 source                  = "artifactory/omega-tf-modules__yahoo-cloud/tf-omega-eks-identity/aws"
 version                 = "~>0.0.20"
 cloud_clusters          = ["sports", "finance", "news", "shared"]
 athenz_domain           = "k8s.omega.aws-nonprod.sample-app-a"
 accountnum              = 9999999999
 services                = [{name = "sample-generic-app", type = "nonprod"}]
}

# GKE identity module
module "omega-gke-identity" {
 source                       = "artifactory/omega-tf-modules__yahoo-cloud/tf-omega-gke-identity/gke"
 version                      = "~>0.0.1"
 project_id                   = "my-gcp-project-id"
 athenz_domain                = "k8s.omega.gcp-nonprod.sample-app-b"
 service_name                 = "sample-generic-app"
 cloud_clusters               = ["identity-nonprod", "data-nonprod", ]
 create_nonprod_environments = true
}
```

Mesh


Istio Mesh
Pod
Pod
Service A
Service B
Data plane
IngressGateway
Proxy
Mesh traffic
Proxy
Ingress traffic
Egress traffic
Discovery
Configuration
Certificates
Control plane
istiod
Pilot
Citadel
Galley

Istio RBAC Automation - Athenz istio Auth

- Use open-source Athenz.io tool for identity
- **Policies** define the RBAC policies for a service
- **Roles** attached to Policy defines identities that can access specified resources
- **Domain** data is synced using k8s-athenz-syncer
- Controller k8s-athenz-istio-auth watches AthenzDomain, Kubernetes services and Istio Authorization policies and updates the cluster config
- Uses SPIFFE ID from X.509 certs, and request principal from JWT tokens, to identify services



Network Policy

- Specification for Pods and Services to communicate with each other at network endpoint level
- Kubernetes Network policy resources control traffic flow at the IP address or port level
- All applications running on Omega on cloud declare NetworkPolicy resources

Network Policy

- Uses Cilium CNI to implement NetworkPolicy resources
- Testing AWS VPC CNI as an alternate
  - https://github.com/aws/aws-network-policy-agent/issues/118
    - Some workloads running on Omega have *a dot* in the name, VPC CNI crashes and does not implement NetworkPolicy
  - https://support.console.aws.amazon.com/support/home#/case/?displayId=170569926701911&language=en
    - Duplicate port section is being added to *PolicyEndpoint*
    - Expose metrics or alerts when NetworkPolicy is not implemented by VPC CNI
- NetworkPolicy rules for applications can be specified as Athenz Micro-Segmentation rules
- Omega team supports a Terraform module to be able to define NetworkPolicy

KubeCon | CloudNativeCon

North America 2024

Key Takeaways

Key Takeaways

- Abstraction for **user to focus on app** instead of infrastructure
- Use of **common tools like KDNC helped us to standardize** the cluster setup end to end using simple interface, based on Terraform
- Bin Packing - Using the right size nodes to **leverage maximum resource utilization** and eventually reduce compute cost
- Use of service mesh and locality-based routing
- **Cluster Security** - use OPA, kube-bench, image policy webhook, Namespace level RBAC etc. to ensure the secure configuration
- **Combine the upgrades**, especially on-prem (OS & K8s upgrade)
- Conduct Well Architected Reviews!

Transformation Continued . . .

- We have 40+ clusters built for transformation for various products
- Successfully launched Header bidding project on GKE, serving ~50K RPS which helped the team to save ⅓ cost
- 2025 is all about migration!

KubeCon | CloudNativeCon

North America 2024

Thank you

KubeCon | CloudNativeCon

North America 2024

Feedback