Microsoft

Configure your AKS Cluster with Confidence

Kendall Roden & Ray Kao, Cloud Native Global Blackbelt

Agenda

Meet your instructors

@kendallroden

@RayKao

Expectations

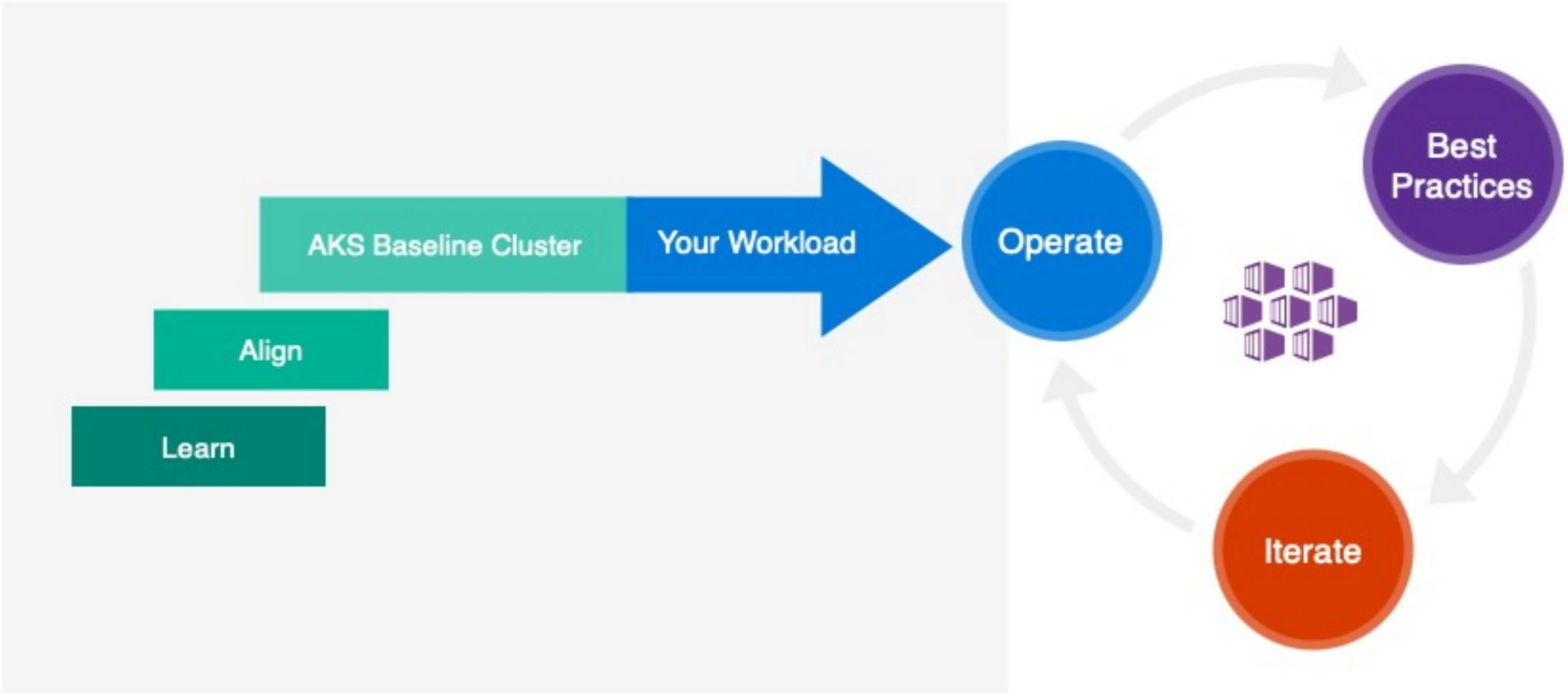
What this session is not:

- An intro to Kubernetes and AKS session
- A guide to all cluster set-up considerations

What this session is:

- An opportunity to fine-tune your approach to AKS configuration
- An opinionated view of AKS best practices for cluster set-up

The Process


AKS Baseline Cluster
Your Workload
Operate
Align
Learn
Best Practices
Iterate

Webinar Series Overview

Today's session
Configure your Cluster with Confidence

04/21/2021 @ 2 EST
Optimize your Cluster for Security and Compliance

04/28/2021 @ 2 EST
Extend your Workload Capabilities

Cluster Provisioning- Infrastructure as Code (IaC)

ARM -> Bicep

- Azure-specific
- Latest Azure resources as released
- Written in JSON
- What-if operation
- Bicep => ARM development language

Terraform

- Cloud/host agnostic
- Azure Terraform provider and modules
- Relies on a state file
- TF validate and plan
- Integrated into Cloud Shell
- Azure Terraform ext. in VS Code

Today's cluster baseline

CloudNativeGBB/webinars(github.com)

```
2021-04-14-configure-your-aks-cluster-with-confidence
    /bicep
    /terraform
    /slide-deck
    README.md (Guide)
```

Cluster Baseline

Compute and Infrastructure

- Managed Identity
- Uptime SLA
- System and User Node Pools

Process

- ACR integration
- Upgrade plan
- Azure AAD + RBAC

Networking

- Azure CNI
- Network Policy with Calico

Observability

- Container monitoring & Log analytics

Deployment overview

Script walkthrough and high-level cluster output

Compute and Infrastructure: Managed Identity

AKS Cluster Azure API Access Approaches

SP or MI is needed to dynamically create and manage other Azure resources i.e., LB or ACR

Service Principal

- Provided to or auto-created by AKS
- Requires registering an app in your AAD tenant
- Valid for year but can be updated/rotated ad hoc
- Should not be used to deploy the cluster itself
- Auto-created SP not deleted on cluster delete

Managed Identity

- Auto-created by AKS for you
- A “wrapper” around a service principal
- Automatically rotated (default => every 46 days)
- Two types: System-assigned and User-assigned

Why use Managed Identity?

With the traditional approach, the Client ID and secret are exposed to both the creator and consumer of the SP

100% identical in functionality and use case between MI and SP - just a reduction in overhead

Couples well with AAD Pod-managed identities (preview)

Compute and Infrastructure: Uptime SLA

Why use Uptime SLA?

Financially backed, 99.**9**5% availability of K8s API Server

Low cost based on # of clusters, not cluster size

Compliance win for critical production workloads

Compute and Infrastructure: Node Pools

Infrastructure- Node Pools

The What

- 100 Nodes per Node Pool (10 pools)
- Single SKU per NP
- User vs. System
- Min/Max pods per node within each NP

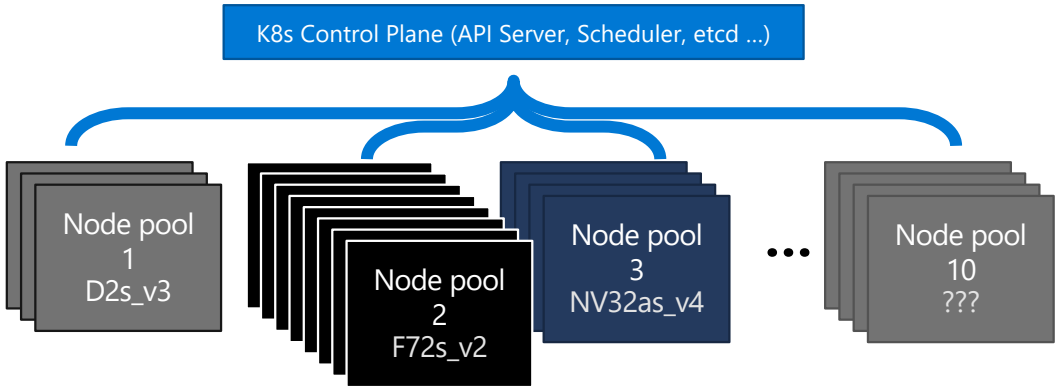
The Why

- Optimize compute needs
- Gain update efficiency
  - Rolling updates per NP
  - Blue/Green NP deployments
- Room for growth and flexibility
- Cluster isolation strategy (logical) - subnet per nodepool

Ramifications- Node Pools

Kubernetes Scheduler Behavior

- **Labels:** `agentpool=nodepoolname`
- **Taints:** `CriticalAddonsOnly=true:NoSchedule`



Additional considerations

- Cross-workload clusters
- Must use Standard SKU LB and VMSS
- NPs cannot span virtual networks
- NP with unique subnet - no network policy support
- A taint can only be set for node pools during node pool creation
- Consider applying Azure Tags to your NP in AKS
- Initial Node Pool profile can't be updated once the cluster is created- must create a separate RM template to update only the node pools

Process:
Upgrade strategy

Decide on a strategy for patching and upgrades

Cluster

Upgrades Control Plane and All Nodes

- Adds new node (aka “buffer node”)
- Cordons and drains old node
- Upgrades old node (new “buffer node”)

Control Plane only

- Update the Control Plane Version only
  - `az aks upgrade --control-plane-only --kubernetes-version 1.19.7`

Node pool only image upgrade

- All node pools
  - `az aks upgrade --node-image-only`
- Specific node pool upgrade
  - `az aks nodepool upgrade --node-image-only`

Considerations

- Major/Minor Releases (N-2 support)
- Security Patches/ updates
- Cluster auto upgrade in preview & node image auto-upgrade in dev

Versioning and upgrade strategy

- Consider using Max Surge (% or Int)
- Make use of Pod Disruption Budgets - but don’t let them block your cluster upgrade process
- Upgrade Control Plane only, then one node pool at a time
- Rolling vs. Blue Green
- Blue/Green Clusters where possible
  - "Chaos" should be the new norm (phrase better)
  - Show the process/workflow thought and considerations to plan for
- Stateful application considerations

YAML

```yaml
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
   name: nginx-pdb
spec:
   minAvailable: 3
   selector:
    matchLabels:
      app: nginx-frontend
```

YAML

```yaml
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
   name: nginx-pdb
spec:
   maxUnavailable: 2
   selector:
    matchLabels:
      app: nginx-frontend
```

Process:
ACR integration and management

ACR Integrations for image security and management

AquaSec and TwistLock integrations

- Build/Registry Scan vs. Runtime Scanning (Webinar #2)
- Walkthrough: Hookup simple scanning

Segregated ACR for environment

- Dev/Test can be most "bloated”
- Prod should be "slimmest" for highly vetted images
- Security polices should be in place to block cluster from accessing anything other than prod ACR for prod env (Admission controllers)

Managed Identity vs. SPNs

- SP Lifecycle mgmt. (manual or automatic)
- RBAC

Image Scanning Setup

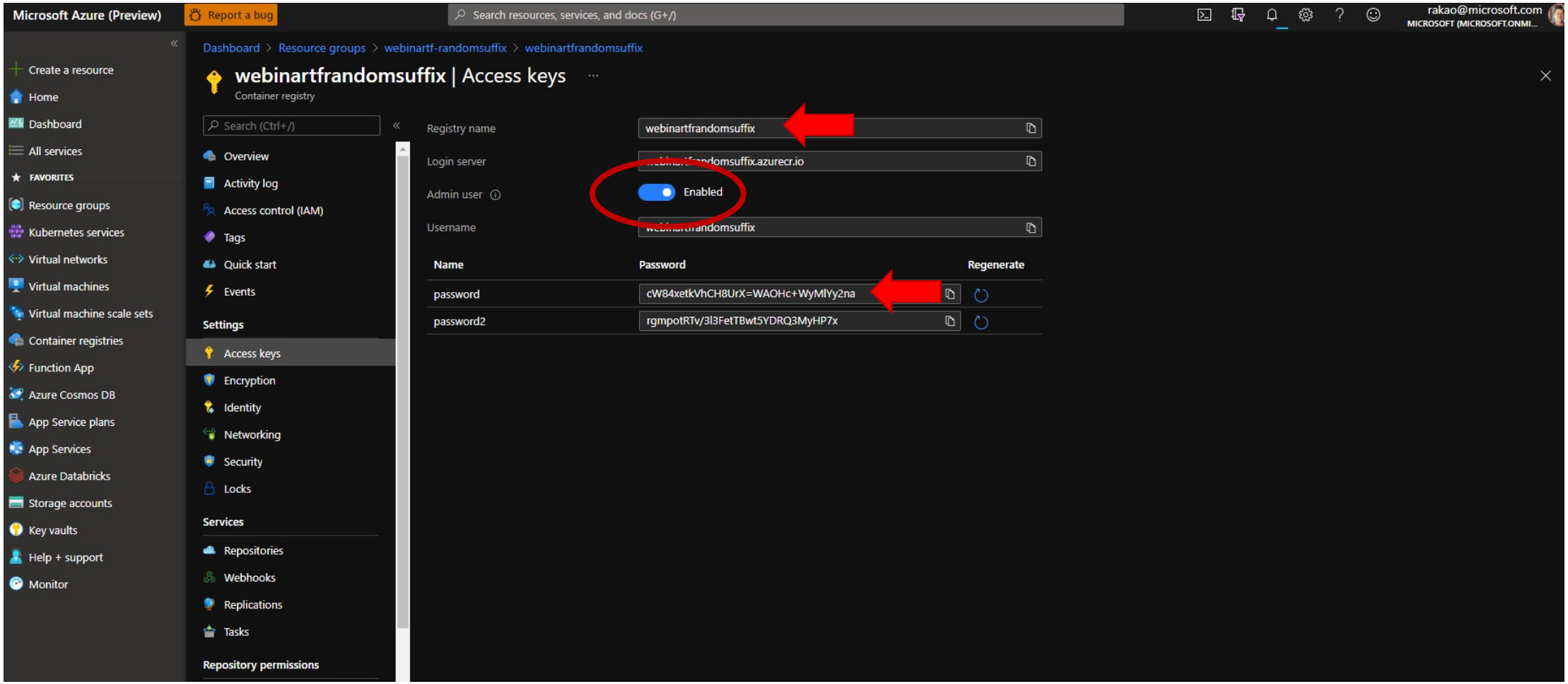


Image Scanning Setup

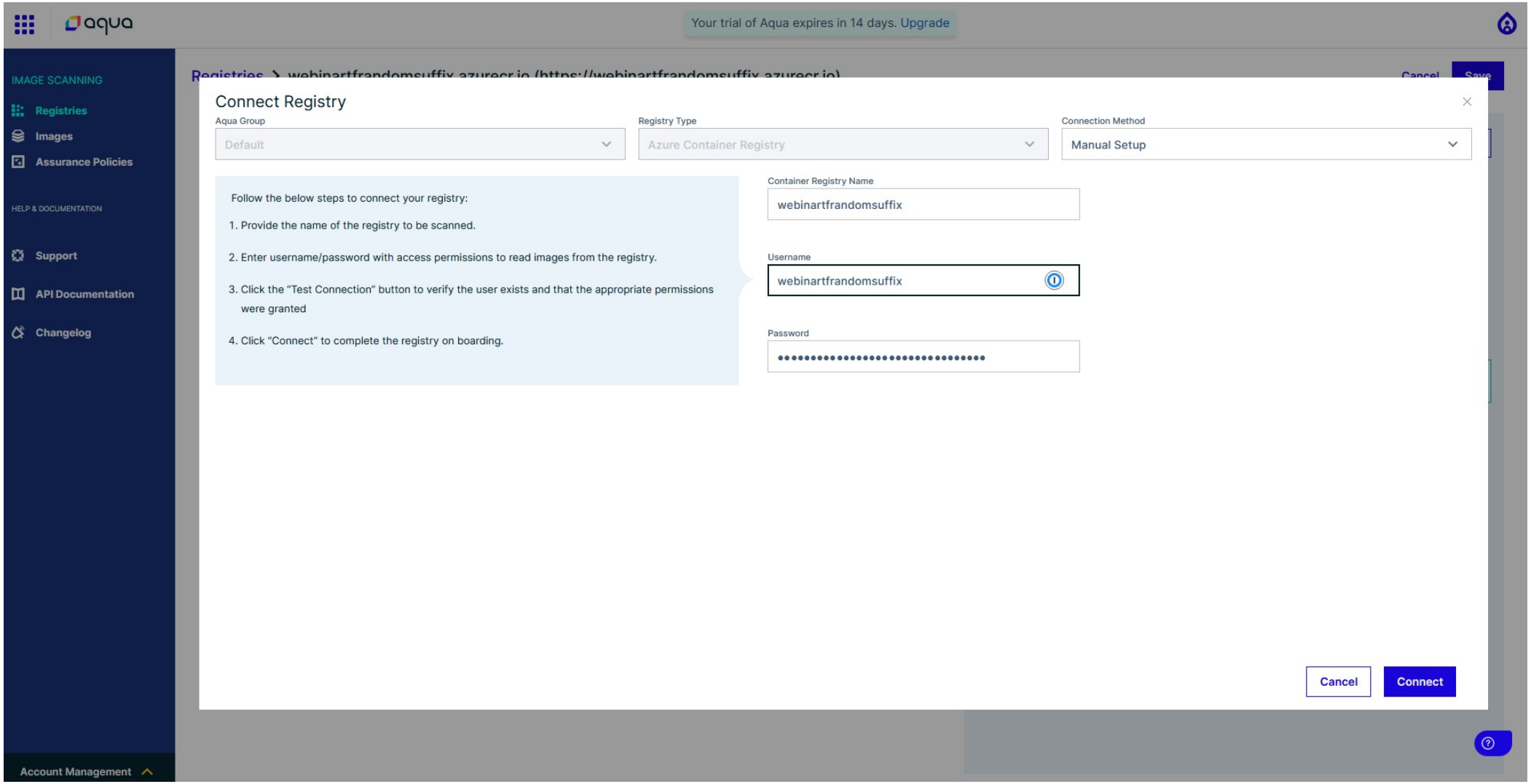


Image Scanning Rules

aqua

Your trial of Aqua expires in 14 days. Upgrade

IMAGE SCANNING

- Registries
- Images
- Assurance Policies

HELP & DOCUMENTATION

- Support
- API Documentation
- Changelog

Account Management

Registries > webinartfrandomsuffix.azurecr.io (https://webinartfrandomsuffix.azurecr.io)

Cancel Save

Registry Name

webinartfrandomsuffix.azurecr.io

Description

Type

Azure Container Registry

Connectivity

Connected

Edit Connection | Test Connection

Aqua Group

Default

Image scanning is not yet enabled for this registry, please enable VS and/or DTA

Vulnerability Scanning (VS)

Dynamic Threat Analysis (DTA)

Enabling DTA will use your plan's units. The number of units will vary based on the number of scanned repositories. Visit the pricing page for more information.

Scan Policy

Scan images which comply with the following pull policies

Include Image Names

E.g. alpine:latest, ubuntu

Exclude Image Names

E.g. alpine:latest, ubuntu

Include Cloud Provider Tag

E.g. env:prod, dep:hq

Exclude Cloud Provider Tag

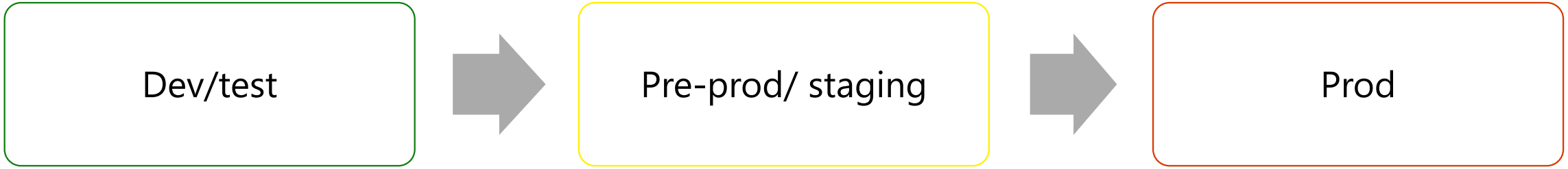
E.g. env:prod, dep:hq

Image Creation Time

All

Retention Policy

Segregated ACR Design

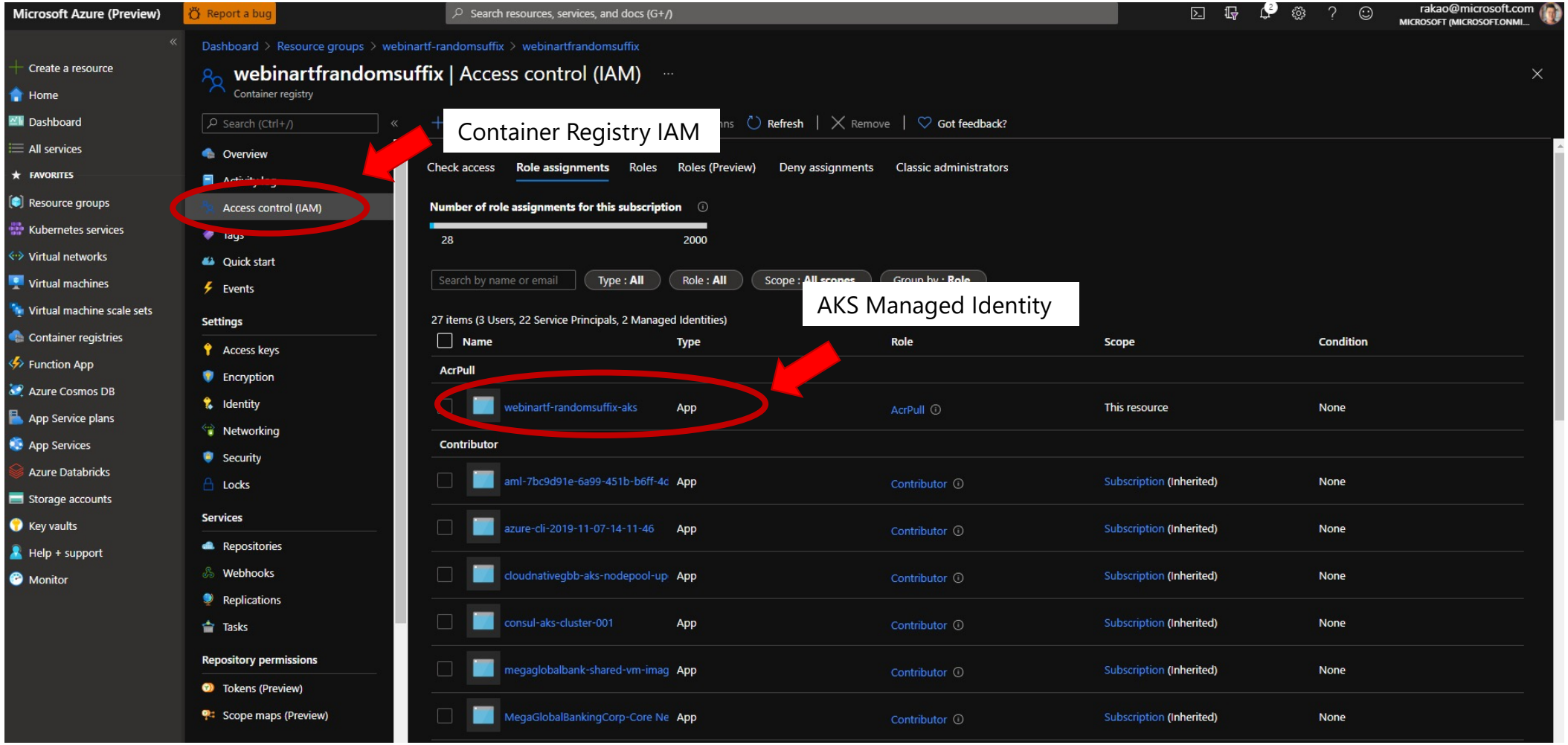


- Can be “basic” tier
- Most user (shared) access
- Stores the most images for use
- Most frequent updates
- Minimal amount of scanning
- Closest to Developers and Test environment

- Can be “basic” tier
- Stores only “release candidate” images
- Semi-Frequent updates
- Moderate amount of scanning

- Can be “premium” tier
- Least user access (Should only be clusters)
- Stores only prod-ready images
- Geo-replicated
- Least frequent updates
- Most frequent updates
- Closest to cluster

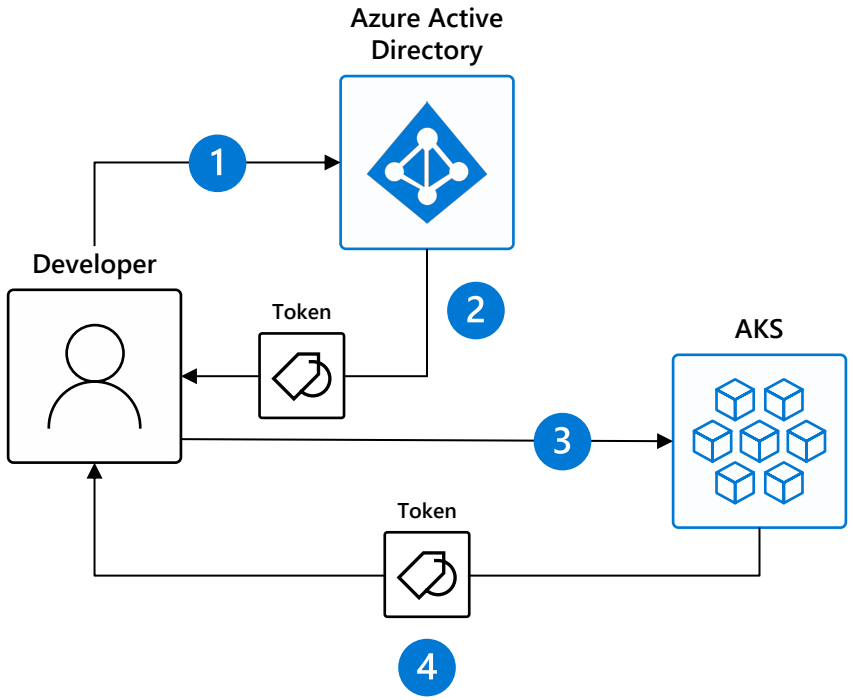
AKS Managed Identity/ACR Access



Process:
Azure AD and Azure/K8s RBAC

Azure AD integration

1. A developer authenticates to the AAD token issuance endpoint and requests an access token
2. The AAD token issuance endpoint issues the access token
3. The access token is used to authenticate to the secured resource
4. Data from the secured resource is returned to the web application



Azure delivers a streamlined identity and access management solution with Azure Active Directory (AAD) and Azure Kubernetes Services (AKS)

Role Based Access Control

- K8s RBAC is designed to work on resources within your cluster
- Azure RBAC is designed to work on resources within your Azure subscription

Authenticating to AKS

Azure AD using (Cluster)RoleBindings

- Requires the Azure Kubernetes User Role
- Not a member of any admin AD group aka. Permissions rely on the K8s roles and rolebindings

Azure AD admin group member

- Requires the Azure Kubernetes User Role
- User is added to one of the cluster admin groups
- AKS auto-generates a Cluster RB that binds to cluster-admin role

Azure AD + Azure RBAC for K8s

- Requires Azure Kubernetes User Role **AND** ¼ AKS RBAC roles or a custom role

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: cluster-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
  - kind: Group
    name: <replace-with-an-aad-group-object-id-for-this-cluster-role-binding>
```

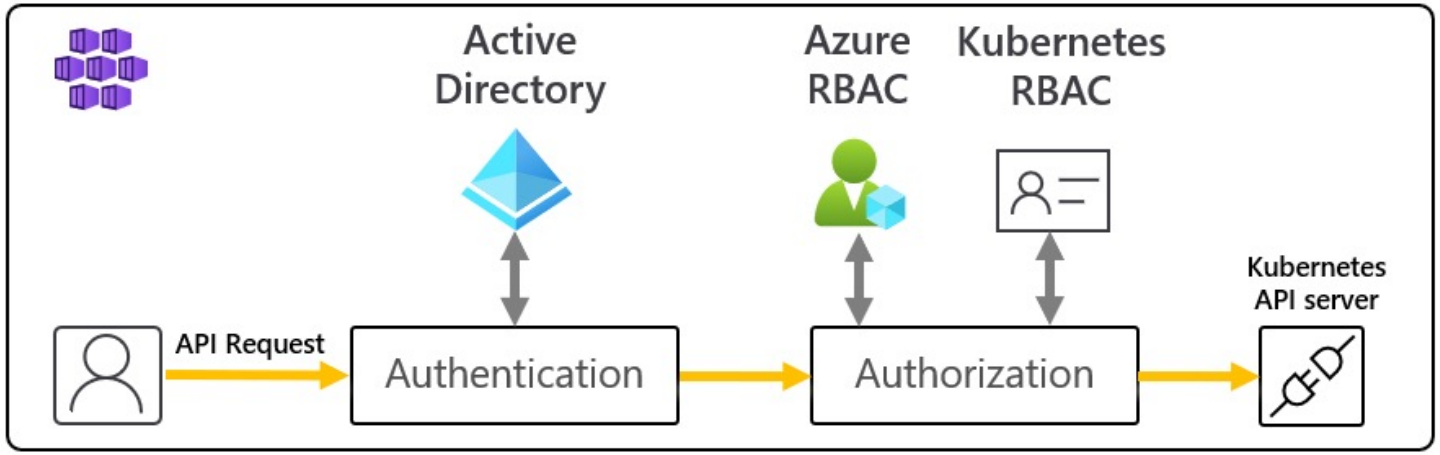
Infrastructure- Azure AD Integration + Azure RBAC

The What

- Use Azure AD to manage cluster access and assign granular permissions
- Four built-in Azure RBAC roles for AKS

The Why

- Secure and streamline
- Extend AD integration with Pod Identity
- No more RB/CRB set-up and management
- Full-cluster access to admin/Ses
- Logical isolation



Considerations

- Azure AD/ RBAC integration may be inhibiting within demo/test/sandbox environments
- Azure RBAC for K8s Authorization is currently in **preview**
- If you do not use this approach, the user responsible for setting up (Cluster)RoleBindings will still have to have a way of gaining access to the cluster, potentially using the legacy admin login if not Azure AD integrated
- Retrofitting RBAC is no fun, set this approach up early on to avoid a headache down the line

Networking:
Azure CNI

AKS Networking Options

Kubenet

- Overlay Network
  - Adds networking complexity and perf degradation
- Only concerned with IP Allocation for each Node (excludes pods in calculation)
- Must maintain Route Tables
- **PRO**: Save on IP Space/Exhaustion

Azure CNI

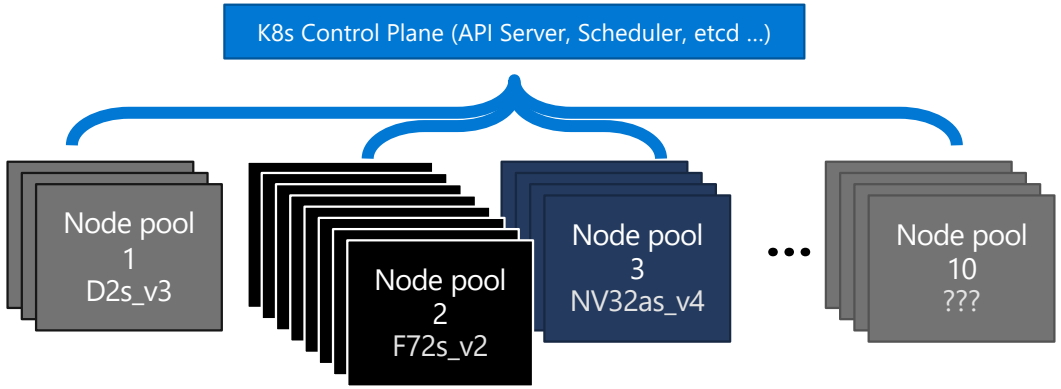
- 1st class citizen – most advancement happens here first
- Better performance (Pods directly addressable)
- No Route Tables
- **CON**: Must pre-size Subnet (see: IP/Subnet Sizing Calculation)

IP/Subnet Sizing Calculation

Remember this: Per node pool min/max pods per node

If:
- max pods per node = 30
- Nodes = 1000
- How many IPs do we need in cluster?

Simple IP Address Calculation:
- 30 pods per Node * 1,000 Nodes = 30,000 IPs
- +1000 Nodes = +1000 Ips
- +1 Node * 30pods (for upgrades)
- +1 Node
- = ~ 31,031 IPs (roughly a /17 subnet with excess)



**Math varies/complex if you change default pods on per node pool basis (NP1 max = 100, NP2 =15 etc.)

Networking:
Network Policy with Calico

Why Network Policy?

All Pods can communicate by default

- Define rules to control the flow of traffic
- Create access policies between Pods
- Protect backend services
- Network isolation of a shared cluster

Two options for Network Policy in AKS

- Azure Network Policies
  - Linux only
  - Azure support offered
- Calico Network Policies
  - First, and most widely deployed implementation of Network Policy across Cloud and On-premises environments
  - Linux and Windows
  - Azure CNI and Kubenet with Linux Nodes
  - Calico community supported
  - Extended policy model

Enable Network Policy now, use later

- Enable policy now, use it later - may not be day 1
- Decide on an approach intentionally upfront and stick with it - retrofitting networking policy != no fun

Observability:
Container insights/ Log Analytics integration

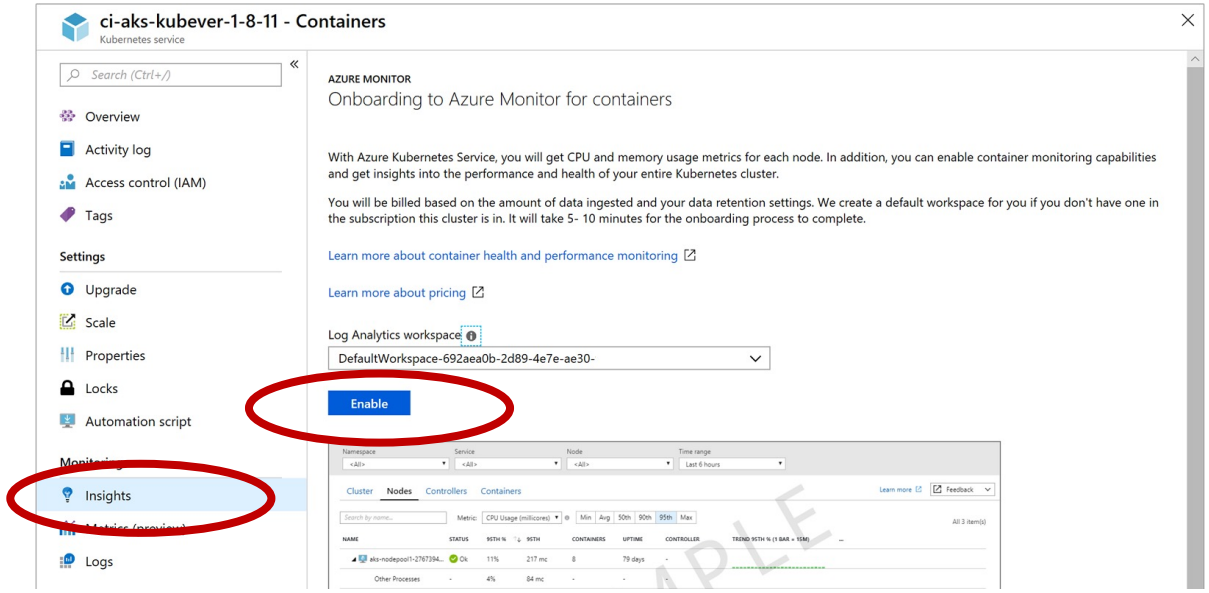
Enabling Azure Monitor for Containers

Bicep Templates/Terraform

```
addonProfiles: {
  omsagent: {
    enabled: true
    config: {
      logAnalyticsWorkspaceResourceID: aksAzureMonitor.id
    }
  }
}
```

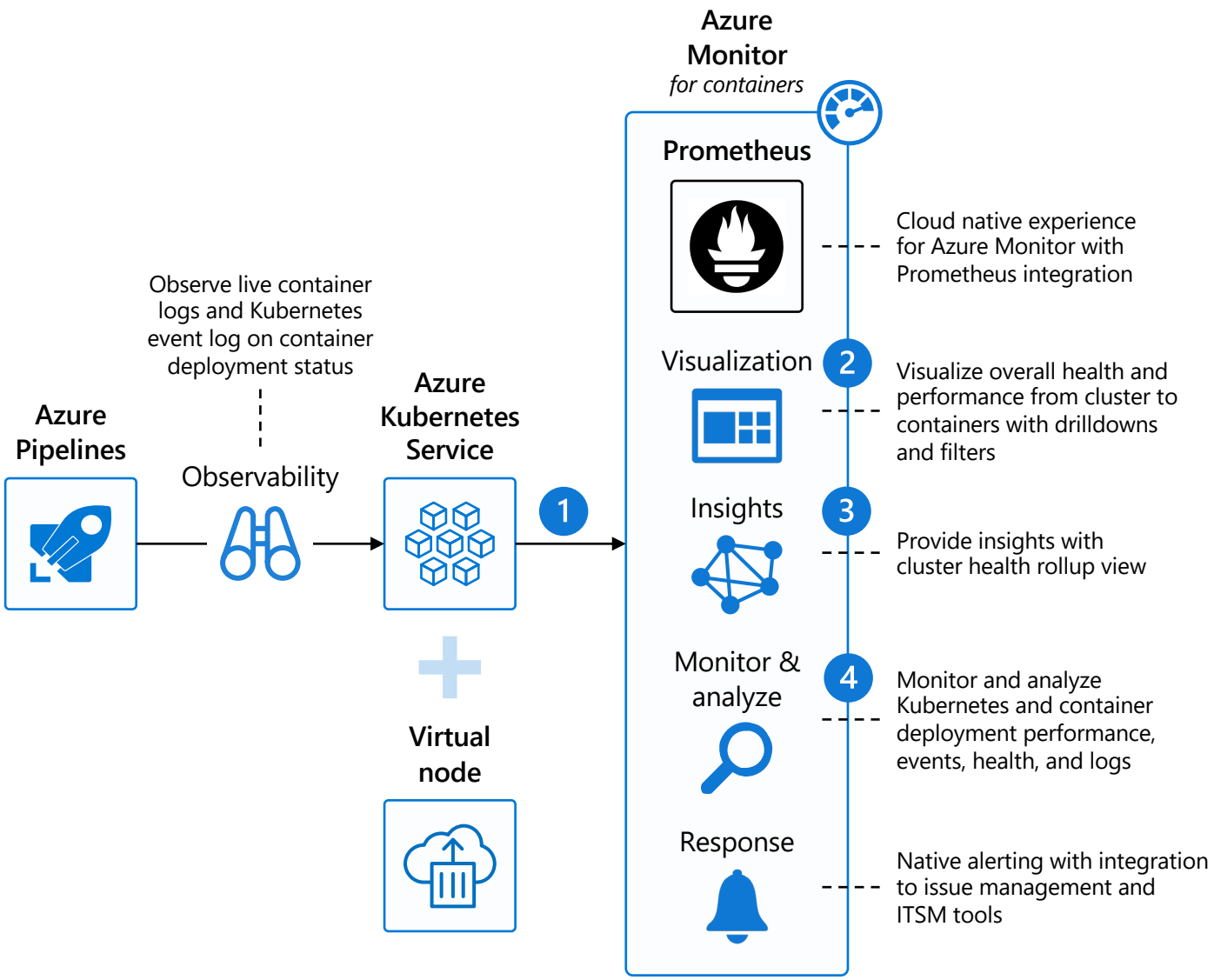
```
addon_profile {
  oms_agent {
    enabled                    = true
    log_analytics_workspace_id = azurerm_log_analytics_workspace.aks.id
  }
}
```

Portal

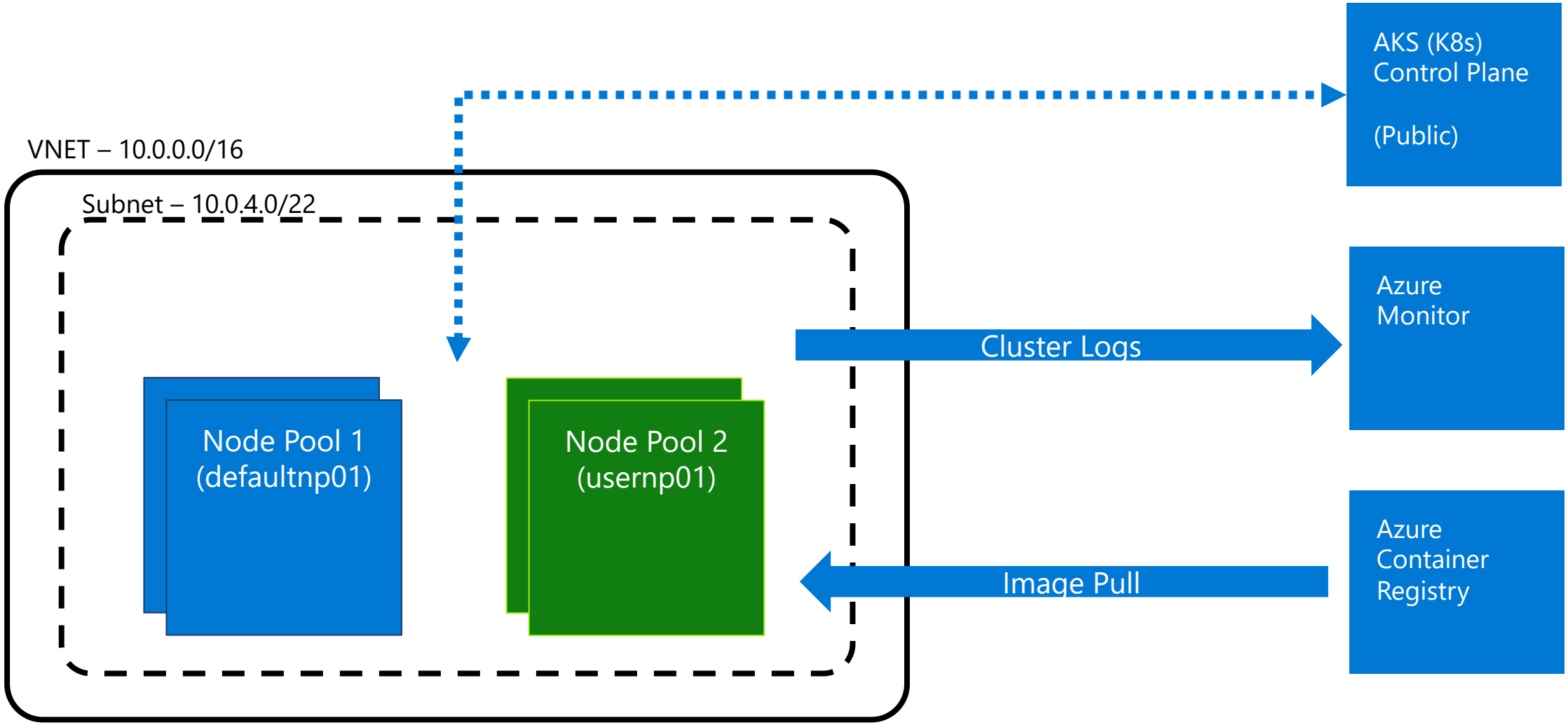


Azure Monitor for containers

1. Get detailed insights about your workloads with Azure Monitor
2. Filter for details about nodes, controllers, and containers
3. See graphical insights about clusters
4. Pull events and logs for detailed activity analysis



Final Cluster (Webinar 1)



Quick Review

Compute and Infrastructure

- Managed Identity
- Uptime SLA
- System and User Node Pools

Process

- ACR integration
- Upgrade plan
- Azure AAD + RBAC

Networking

- Azure CNI
- Network Policy with Calico

Observability

- Container monitoring & Log analytics

Questions about our setup…

- How “secure” is this?
- Is HTTPS communication over “public” internet, ok?
  - Stays within Azure Network but is not a “private link”
- How can we lock it down a bit more?
- How do we meet requirements if there are compliance/regulation requirements?

Join us for the next session on April 21st

Additional Q&A

Join us for AKS Office Hours!

Hosted by the Cloud Native GBB Team every other Thursday from 11-12 CST!

- Provide AKS customers with updates pertaining to AKS and the Cloud Native Ecosystem
- Host a short talk and/or demo on Cloud Native technologies related to Kubernetes and AKS
- Collect feedback from customers on issues, blockers, use cases, and questions related to AKS

Other Resources

AKS Public Office Hours

https://aka.ms/akspublicofficehours

Microsoft Cloud Native GBB YouTube Channel:

https://www.youtube.com/channel/UCvdABD6_HuCG_to6kVprdjQ

Kubernetes Learning Path:

https://azure.microsoft.com/en-us/resources/kubernetes-learning-path/

AKS Checklist:

https://www.the-aks-checklist.com

AKS Solution Journey

https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/containers/aks-start-here

AKS Workshop (MS Learn):

https://docs.microsoft.com/en-us/learn/modules/aks-workshop/

GBB AKS Secure Workshop:

https://github.com/CloudNativeGBB/aks-secure-workshop