bluu

Cloud-native applications on Azure Kubernetes Service: the bigger picture

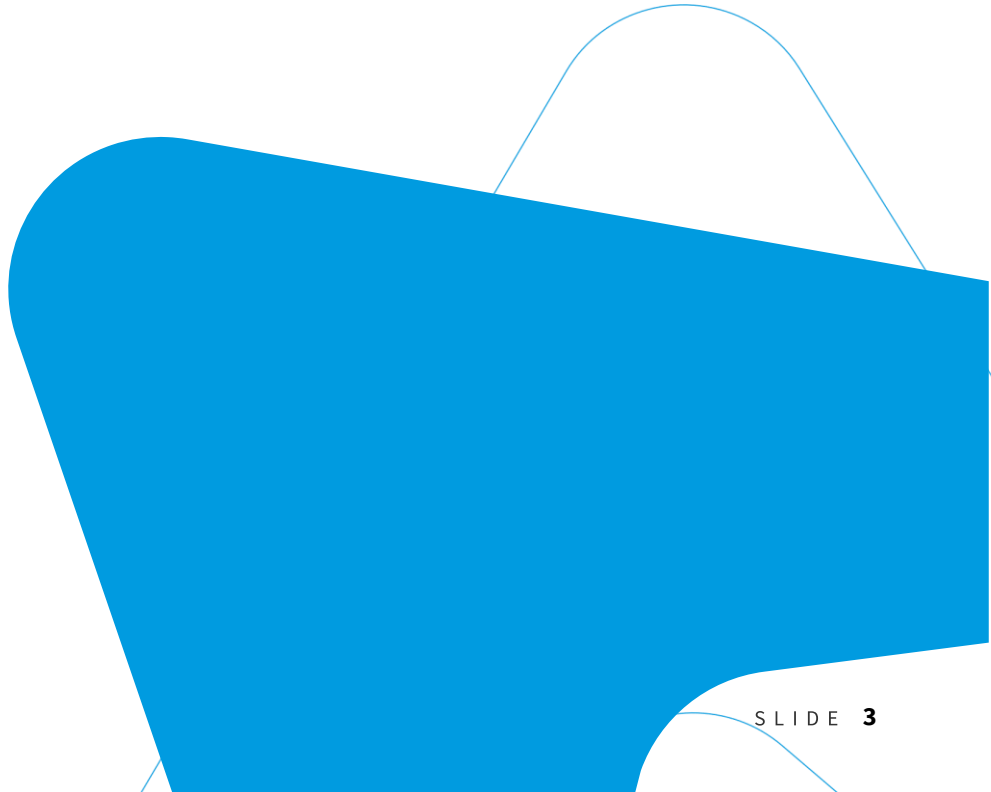
Who Am I?

Geert Baeke

@geertbaeke

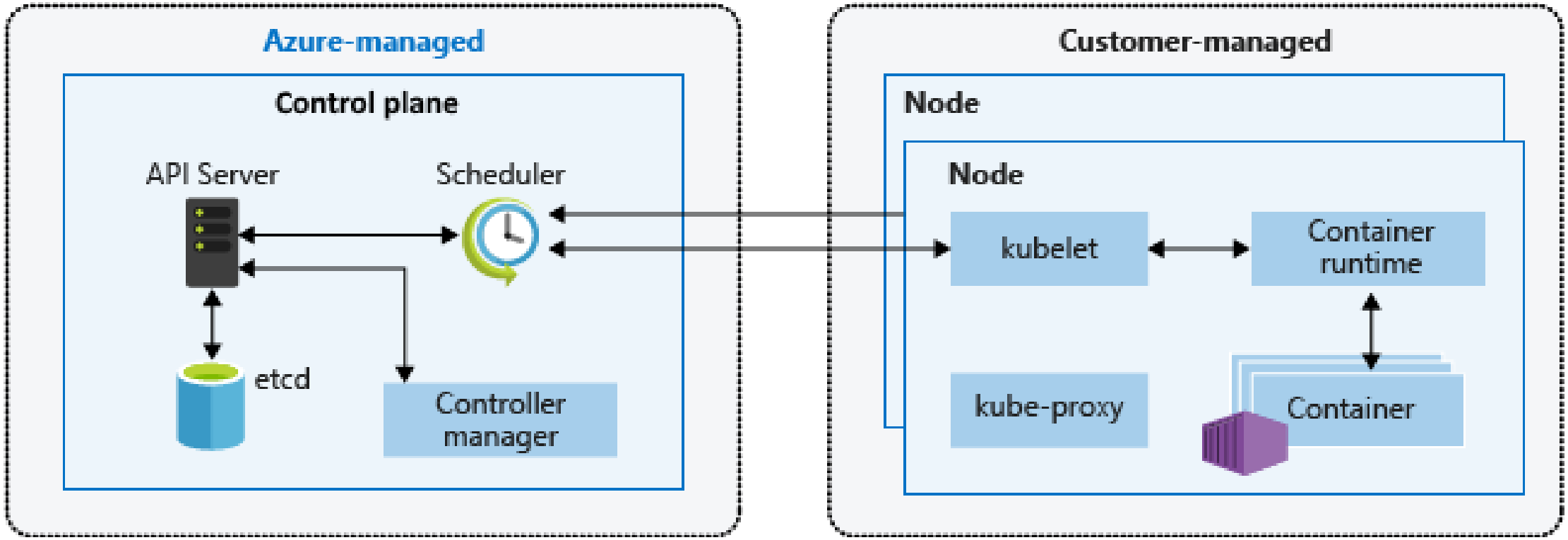
https://blog.baeke.info

https://github.com/gbaeke



Just enough Kubernetes

Azure Kubernetes Service (AKS)



AKS is not PaaS!

- You have full control over the worker nodes
- You need to reboot the worker nodes after patching
- You need to upgrade the cluster


YAML

```yaml
# Made with 💔 by https://twitter.com/geoffreyhuntley after one too many CustomResourceDefinitions
# Improvements welcome, mash the ↗️ Octocat ↗️ and share how YAML makes your life better.

No:
  Body:
    Wants:
      To:
        Write:
          - YAML

# 🤔 Why YAML is the right devops technology for you 🤔
#
# - 100% test coverage, always compiles just fine with no errors or warnings, always shippable
# - no enforced error handling during development because runtime "panic at the disco" in production is dope
# - "something broke" is way better than stack traces with line numbers
# - you need to burn hours as part of setting up a new CI pipeline
# - safe choice with unquestionable industry adoption, "used by kubernetes"
# - is marginally better than windows.ini
# - unlike json [1], YAML supports comments
# - you need a super safe way to "execute this code"

# 🏮 wait a sec, did you say "executable yaml"?? 🏮
# - https://ruby-doc.org/stdlib-2.4.0/libdoc/yaml/rdoc/YAML.html#module-YAML-label-Security
# - https://www.php.net/manual/en/function.yaml-parse.php#refsect1-function.yaml-parse-notes
# - https://lgtm.com/blog/swagger_snakeyaml_CVE-2017-1000207_CVE-2017-1000208
# - https://github.com/yaml/pyyaml/wiki/PyYAML-yaml.load(input)-Deprecation

# 🚨 Anyone who uses YAML long enough will eventually get burned when attempting to abbreviate Norway 🚨
```

NOYAML.COM

Demo

Running CSharpWars on AKS

Deploying your apps

Imperative

Run commands

Interactively or scripted

Declarative

Declare the desired state

Back-end system sets the state

Combine declarative management with automation

Use traditional deployment pipelines

E.g. Azure DevOps

Use GitOps

E.g. Flux, Argo, …

Likely a combination

Traditional CD Systems

Pipeline driven by a CI/CD system (such as Azure DevOps)

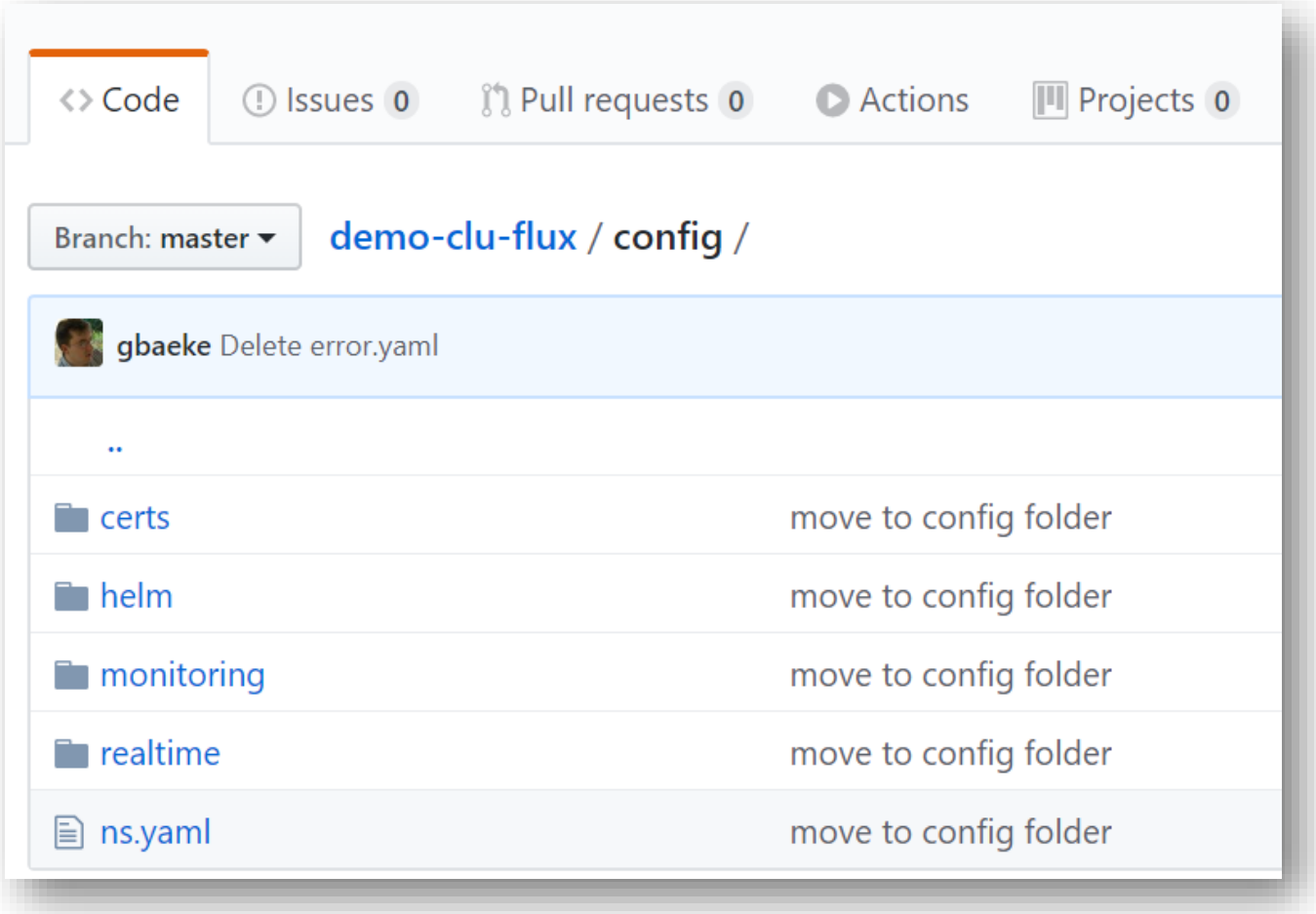
Steps that execute tasks

CI/CD systems need credentials to your clusters

Push-based

GitOps

- A method of continuous deployment (CD)
- Git is the single source of thruth for declarative infrastructure and applications
  - Revisions
  - Change control
  - Rollback
- Pull-based
- No need to provide cluster credentials to external system

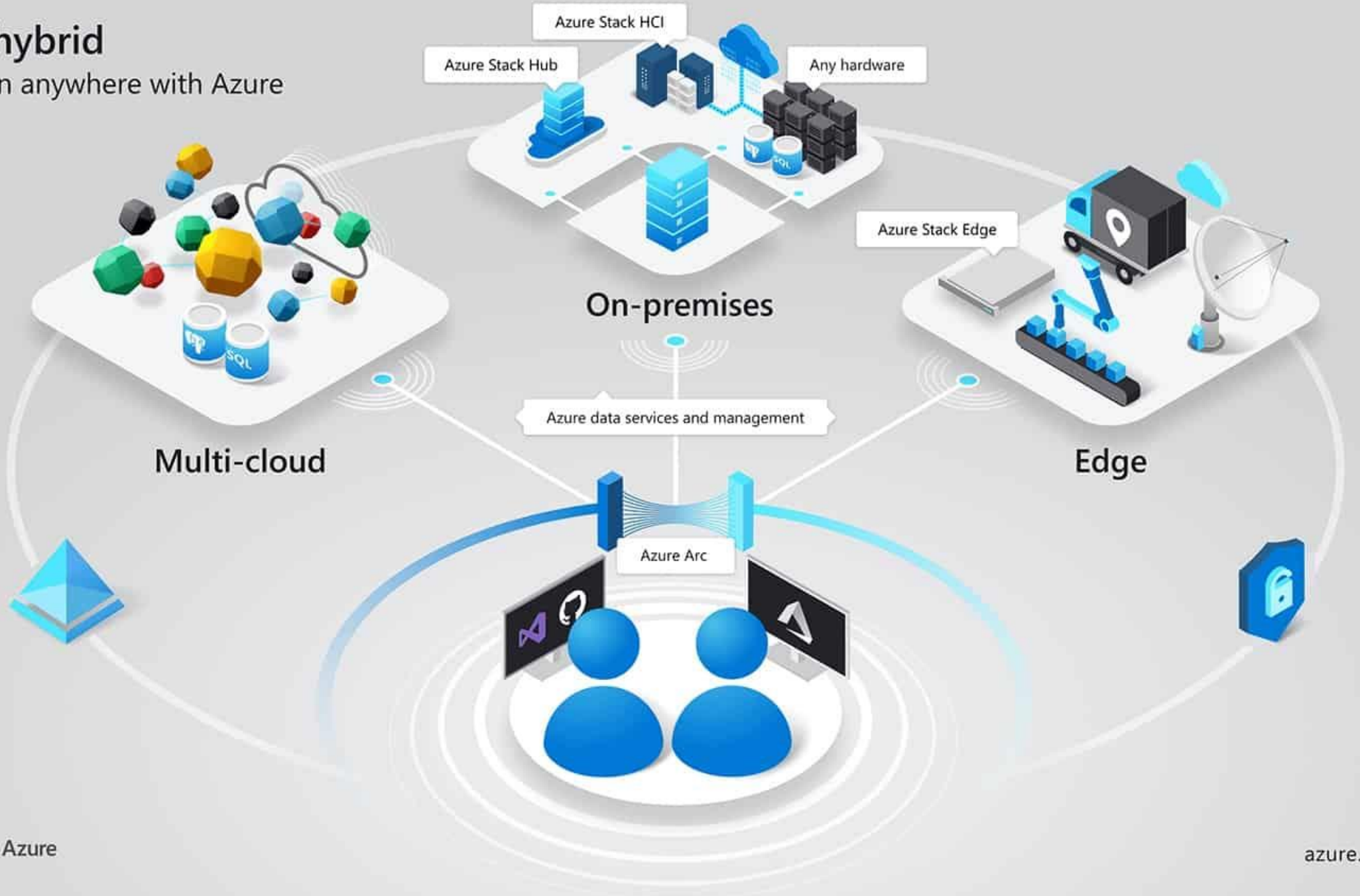


GitOps Operators

- Act on configuration files in a git repository
- Continuously running (e.g. check every minute)
  - Or triggered by web hook
- Additional features: pruning, image updates, …
- Examples
  - Flux
  - Argo

Azure hybrid

Innovation anywhere with Azure



Microsoft Azure

azure.com/hybrid


BUZZWORDS
BUZZWORDS EVERYWHERE

Demo

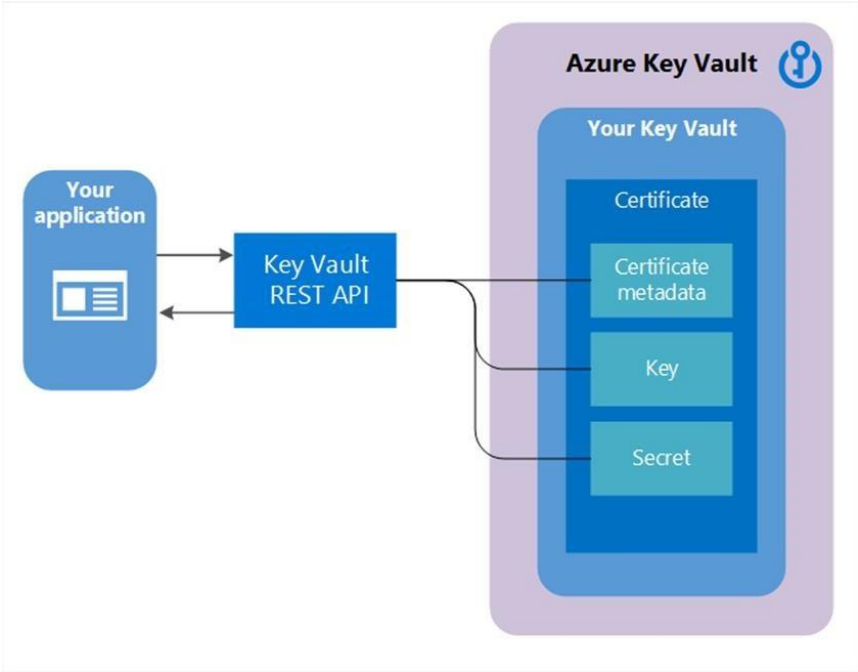
Deploying CSharpWars using Flux

Secrets


FACT:
GIT WAS NOT DESIGNED
TO BE A SECRET STORE
imgflip.com

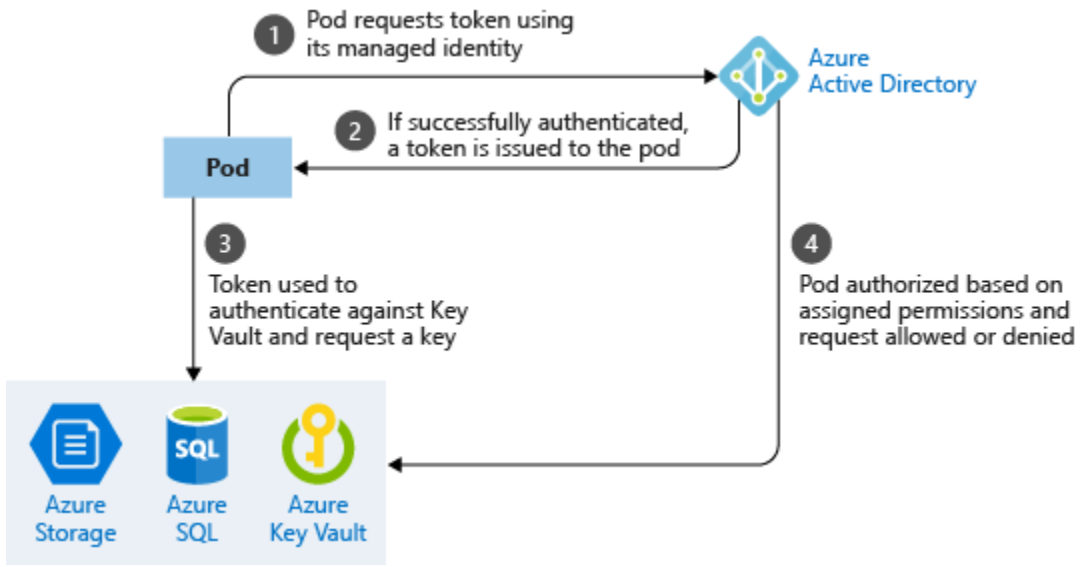
Getting Secrets from Azure Key Vault

- Directly by your application
- Use an operator like **Azure Key Vault** to Kubernetes
- Use **Kubernetes-keyvault-flexvol**
  - https://github.com/Azure/kubernetes-keyvault-flexvol



Pod Identity

- Allow a pod to use Managed Identity
- Allow the Managed Identity to access Key Vault
- No need to configure your application with Key Vault credentials
- Just provide the Key Vault URI via a ConfigMap



Azure Key Vault to Kubernetes

- See https://akv2k8s.io/
- Two components:
  - Controller
  - Injector
- Can authenticate to Key Vault with the AKS security principal
- Controller creates regular K8S secrets

```yaml
apiVersion: spv.no/v1alpha1
kind: AzureKeyVaultSecret
metadata:
  name: eventendpoint
  namespace: default
spec:
  vault:
    name: gebakv
    object:
      name: EventEndpoint
      type: secret
  output:
    secret:
      name: kedasample-event
      dataKey: EventEndpoint
      type: opaque
```

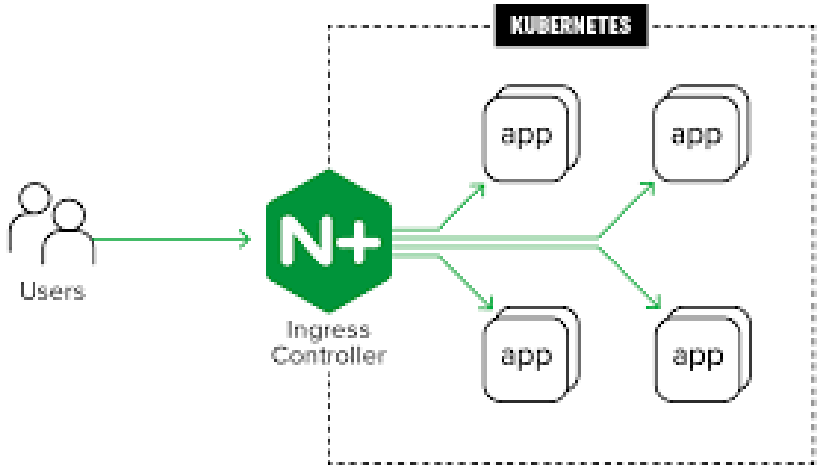
Demo

Configuration of CSharpWars

Exposing services securely

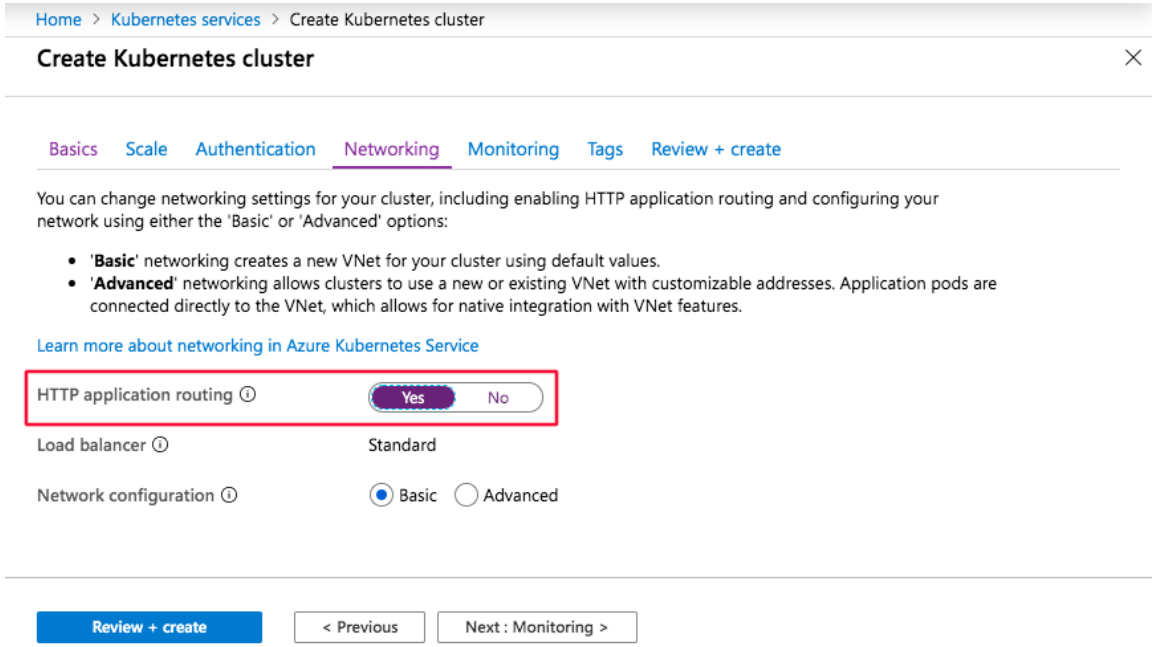
Ingress Controllers

- Basically a **reverse proxy**
- Exposes Kubernetes services
  - Inside Azure Virtual Network: via internal load balancer
  - To the Internet: via public load balancer
- Configured via Ingress objects
- Many Ingress Controller to choose from
  - nginx, nginx+, Traefik, Voyager, Azure Application Gateway, …



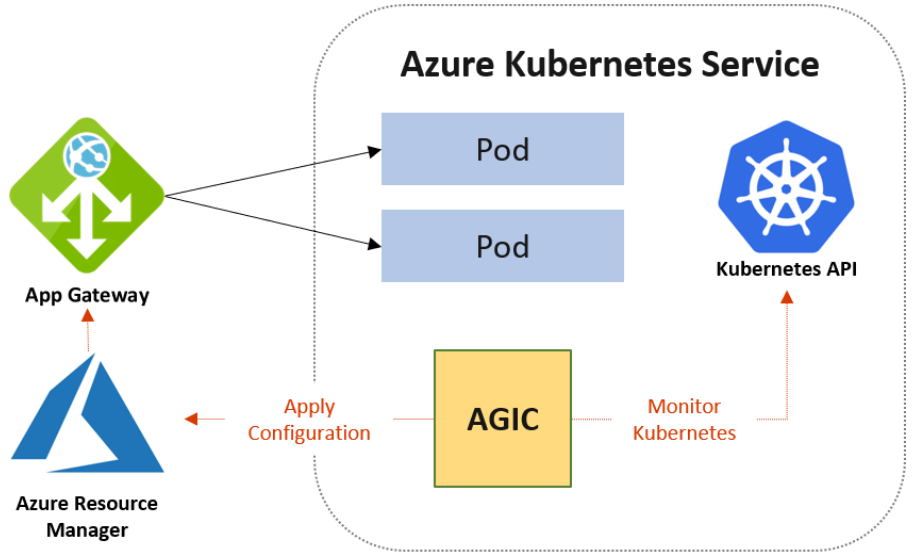
What about AKS HTTP Application Routing?

- It’s an Ingress Controller
- Not to be used in production



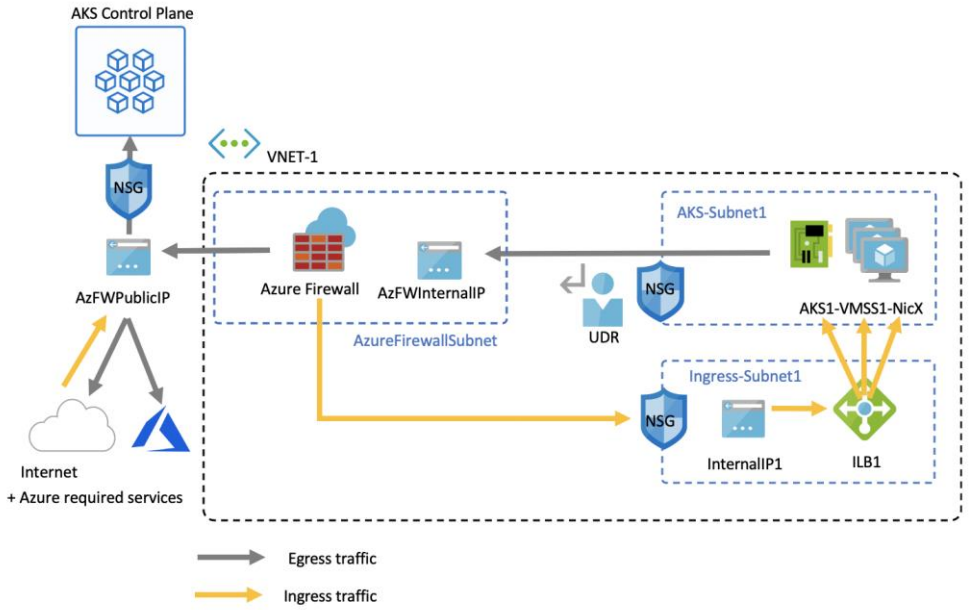
Application Gateway for Ingress

- Application Gateway Ingress Controller runs on K8S cluster
- Monitors Ingress definitions and configures App Gateway
- App Gateway routes directly to the IP addresses of the pods



AKS and Azure Firewall

- Restrict external access by AKS from the AKS subnet in your virtual network
- Use Azure Firewall to control access to specific URLs
  - Follow https://docs.microsoft.com/en-us/azure/aks/limit-egress-traffic rigorously
- Route ingress traffic via a DNAT rule to an internal ingress controller
- Use Network Policies for internal cluster traffic



Augment with Azure Front Door

- Azure Front Door as the entry point to your application
- Adds CDN, caching, security via WAF rules
- Requires publicly exposed back-end with valid certificates
  - Restrict access to Front Door via whitelist
  - Use Let's Encrypt certificates with cert-manager
- Use Microsoft-provided certificates for the front-end

Demo

Ingress for CSharpWars with ngnix, cert-manager and Azure Front Door

Tips

- Configure whitelist:
  - ingress.kubernetes.io/whitelist-source-range: W.X.Y.Z/..
- Use DNS verification for Let's Encrypt
  - cert-manager supports most DNS providers (e.g. Azure, CloudFlare, …)
- Use "origin" URLs such as csharpwars-o.baeke.info for your backend
- Set affinity rules at Front Door and Ingress level
  - nginx.ingress.kubernetes.io/affinity: cookie

Conclusion

- Automate “all the things” with a declarative data driven approach
  - Validate your YAML and Helm charts
- Use separate secret stores and integrate with Kubernetes
- Know your Ingress Controllers
  - Use cloud-provided Ingress Controllers if possible
- Combine with Front Door for global caching and Web Application Firewall

Additional Information

- Deploy AKS with useful add-ons using Azure DevOps:

  https://blog.baeke.info/2019/12/06/deploy-aks-with-nginx-external-dns-helm-operator-and-flux/

- GitOps and Flux
  - https://blog.baeke.info/2019/09/17/gitops-with-weaveworks-flux/
  - https://blog.baeke.info/2019/10/10/gitops-with-weaveworks-flux-installing-and-updating-applications/