weaveworks

CASE STUDY

INTRODUCTION

Wealth Wizards, an online financial planning and advice company in the UK, provides a white-label SaaS platform that combines chartered financial planning, actuarial science, and smart software technology to deliver expert advice at an affordable cost. Their drive towards automation reduces the time required for financial advice from weeks to a matter of hours. Their scalable, secure, and robust cloud-based applications offer advice and guidance from personal to employer-level financial services, including pensions, retirement, debt, mortgages, and more. Wealth Wizards believes that everyone deserves the opportunity to create a positive financial future.

CHALLENGES

For Wealth Wizards it is extremely important to quickly discover and eliminate code vulnerabilities as well as have full visibility of their cloud environment. Auditors need to have the confidence that Wealth Wizards can analyze any breach or data loss at all times. Wealth Wizards wanted to run a dynamic, cloud-native environment where workloads can run anywhere, and also wanted to control traffic through access restrictions.

> We're here to put optimism back into financial advice through the use of smart technology and financial intelligence so that people can feel in control of their money."

Wealth Wizards currently run their Kubernetes containerized infrastructure on AWS - a multi-AZ deployment in the EU regions. Terraform is used to implement a typical PCI compliant, 3-Tier, 3-AZ VPC for their microservice platform along with the high-level Security Groups. They use KOPS to manage the provisioning and rollout of the clusters and use a suite of custom in-house tools to manage the deployment and configuration of microservices across the kube clusters.

Wealth Wizards chose to use AWS due to their wide support within the community as well as the huge range of products AWS offer to make cloud computing easier, safer and more secure. "When you're working with large financial institutions who use the cloud, it makes sense to use a provider everyone understands and is familiar with." Richard Marshall, Head of Platform, Wealth Wizards

Wealth Wizards

HIGHLIGHTS

- Allowed Wealth Wizards to secure microservices running in Kubernetes on AWS for security and compliance auditors

CHALLENGES

- Control ingress traffic between microservices
- Discover and eliminate code vulnerabilities quickly
- Get full visibility on data breaches and data losses

SOLUTION

- Implement Weave Net to enforce flexible and powerful network policies to secure microservices in Kubernetes

CONTACT US

www.weave.works

help@weave.works

Wealth Wizards have been using a feature in Kubernetes called Network Policies for a little while. Network Policy allows them to control traffic through access restrictions (similar to a virtual firewall). Network Policies are application-oriented and can evolve with the application.

Network policies are managed through the Kubernetes API but a component is required that enforces the policies. Weave Net is a network controller that enforces the policies.

SOLUTION

Weave Net creates a virtual network that connects Docker containers across multiple hosts and enables their automatic discovery. If you're using Weave Net as the network pod layer, it comes bundled with a network policy controller, which manages and enforces the rules you set up in Kubernetes. Weave Net is easy to install with a one-command setup. Network policy rules are driven from the Kubernetes control plane with no additional components to install.

> "Weave Net gives us the ability to drive security right down to the microservices layer with none of the cost of micromanaging the services. It also allows us to extend the security outside the boundaries of the Kubernetes cluster and include the legacy components. A double win!"

Wealth Wizards needed a solution for securing their Kubernetes clusters on AWS for security and compliance for their auditors. Using Weave Net for network policies made it easy to link workloads and security policy within their Kubernetes cluster.

Network policies are very flexible and powerful. As an example, let's say we had a workload in our cluster and we wanted to stop any other workload in the cluster talking to it. We can do this by creating a network policy that restricts access, and only allows ingress to it via the ingress controller on a specific port. Network policies gives you a lot of flexibility for securing your clusters, and it comes at very little cost. Besides using namespaces, you can define very fine-grain policies with label expressions.

So what does a network policy look like at Wealth Wizards?

The following policy will prevent ingress access into a namespace to everything except traffic originating in the namespace:

```
___
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: namespace-isolation-live
  namespace: live
spec:
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          name: live
  podSelector: {}
_______
```

The above policy says:

Allow ingress from resources where the namespace matches "live" and apply to all pods (the '{}' ; denotes ALL pods in the namespace).

In order to actually serve traffic from the namespace we have another policy:

```
___
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: elb-isolation-live
  namespace: live
spec:
  ingress:
  - ports:
    - port: 443
      protocol: TCP
    - port: 80


     protocol: TCP


 - {}
  podSelector:
    matchLabels:


     name: ingress-controller
___
```

Allow ingress from all pods, where the ports are `TCP:443` and `TCP:80` and apply the policy to pods with have the label matching: ingress-controller (which the Wealth Wizard ingress controllers do).

An area to bear in mind is that network policies are implemented using iptables which means that they only operate at the IP and TCP level. Network policies are not able to operate on URLs (e.g. layer 7), so you need to build rules into an ingress controller or look at using a service-mesh like Istio. Combining both of these will give you a greater level of security with two independent but complimentary systems which manage access to resources.

RESULTS

"We deployed the Weave agent onto each Kubernetes node. It manages things like inter-pod routing, but because it's installed at the OS layer, it also has access to manipulate the iptables rules. This is how it implements the access restrictions defined by the network policies. Each policy is converted to a collection of iptables rules, coordinated across each machine, which translates the Kubernetes tags into something that can be recognized on each machine." said Richard Marshall, Head of Platform at Wealth Wizards.

For more information about how Weave Net works, read our guide.

KEY BENEFITS

- **Allowed Wealth Wizards to extend security outside of the Kubernetes cluster and include legacy components.**
- **Weave Net was easily installed with a one-command setup.**
- **Enabled Wealth Wizards to create a the network pod layer that comes bundled with a network policy controller, which manages and enforces the rules you set up in Kubernetes.**

ABOUT WEAVEWORKS

Weaveworks makes it fast and simple for developers and DevOps teams to build and operate powerful containerized applications. The Weave Cloud operations-as-a-service platform provides a continuous delivery pipeline for building and operating applications, letting teams connect, monitor and manage microservices and containers on any server or public cloud. Weaveworks also contributes to several open source projects, including Weave Scope, Weave Cortex and Weave Flux. It was one of the first members of the Cloud Native Computing Foundation. Founded in 2014, the Company is backed by Google Ventures and Accel Partners.

Visit us at www.weave.works.