EASY Kube

PDF

COMMON KUBERNETES

SECURITY MISCONCEPTIONS AND MYTHS

In this article we are discussing the Top 10 security misconceptions about Kubernetes. Did you know, that namespaces do not isolate pods? Did you encrypt your secrets? Can someone listen to my traffic between nodes? All this and more is answered in this article.

CONTACT

+1 (650) 382 0775

+49 160 1136770

easycloudhost@datafortress.cloud

easycloudhost.de/services/easykube/

KUBERNETES

SECURITY MISCONCEPTIONS AND MYTHS

Kubernetes can be complex, but once you get to know the basics, it is a wonderful tool allowing you to get up and running quickly.
Once you have deployed your first services, there are common pitfalls in terms of security that you might miss.

This article touches on the Top 7 misconceptions.

Author: Justin Güse, CEO
EasyCloudHost.de &
DataFortress.cloud

1. KUBERNETES NAMESPACES ISOLATE CONTAINERS

Thinking that namespaces isolate containers is a common misconception. They are only showing up in that specific namespace, but are not isolated from each other. Let us say you are running a postgresql deployment in the "default" namespace as service name "psql-deployment". If you start up another container, you are able to connect to it connecting to "psql-deployment:5432". Now, if you move your second deployment to another namespace, you will not be able to connect anymore.

Now it might be easy to think that namespaces isolate network resources, but the truth is only the routing changed, and the pod can still communicate with the psql-deployment. We could try this out by using the internal DNS routing called "psql-deployment.default.svc.cluster.local". As you can see the structure is [SERVICE].[NAMESPACE].svc.cluster.local, and every service can be accessed with that.

This poses a huge security risk but can be avoided by using special networking plugins in Kubernetes. Common plugins are Calico, Flannel, Weave, and Cilium. They can be set up to isolate namespaces, and containers, and specify different Security Groups and Access Control Lists as you might know from other cloud providers.

CONTACT

☎ +1 (650) 382 0775

☎ +49 160 1136770

✉ easycloudhost@datafortress.cloud

🌐 easycloudhost.de/services/easykube/

EASY Kube
Managed Kubernetes

KUBERNETES

SECURITY MISCONCEPTIONS AND MYTHS

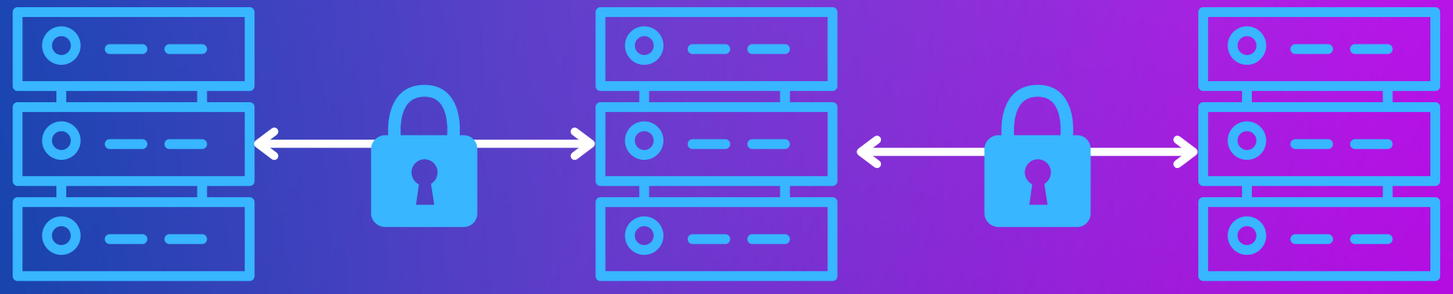
2. INTER-NODE COMMUNICATION IN KUBERNETES IS ENCRYPTED

A similar issue happens with the communication between Kubernetes nodes. Let us say you are encrypting your WordPress (or any other) traffic with automated Cert-Manager and Letsencrypt. All good, right? Wrong - because traffic in between nodes is unencrypted by default.

But what does that mean?

If one node is located in Frankfurt, and another in Munich for availability reasons (good on you!), the communication between nodes still happens unencrypted. This means, that anyone gaining access to the data center connection could read your traffic!
Of course, this is quite strongly prevented if your hosting provider is reliable, but especially if you are using self-hosted setups you should be careful in doing this.
And what can you do against that? You could either manually use self-signed certificates for the pods, or - easier - use Kubernetes plugins for that. The previously mentioned networking plugins offer a lot of functionality.

The plugins implement a so-called "service mash", which deploys a sidecar along with your pod, that is taking care of TLS termination with mTLS enabled between pods.
Calico does not implement this by default, but Linkerd and Istio can use traffic encryption in combination with Calico.
Linkerd is the easiest plugin in these regards, as it implements mTLS out of the box.



CONTACT

+1 (650) 382 0775

+49 160 1136770

easycloudhost@datafortress.cloud

easycloudhost.de/services/easykube/

EASY Kube
Managed Kubernetes

KUBERNETES

SECURITY MISCONCEPTIONS AND MYTHS

3. VELERO BACKUPS ARE SECURED AND ENCRYPTED

Say you have Backups implemented, and save them to AWS S3 for safekeeping. A good step for sure in terms of disaster recovery, but did you know, that Velero does not encrypt your backups by default?

This can have serious implications if you are running an isolated self-hosted Kubernetes cluster because you are from the finance industry or store medical data, but the backup is stored in true form on an American cloud provider.
Many European countries specify to not use American cloud providers due to security risks involved, and many still keep on using S3 for backups. If someone accesses these object storages in the public cloud, there is no encryption on the files stored.
Unfortunately, at the time of writing, there is no Backup service offering encryption. My solution was to create custom Kubernetes Jobs, that dump the data, encrypt it in the Job, and then store it in object storage provider interfaces.

4. KUBERNETES SECRETS ARE ENCRYPTED

You might have created secrets in Kubernetes, and might have taken a look at them in your running cluster. "aWxvdmV0ZW5kaWVz", wow that looks quite encrypted! But again the hard truth is: Wrong!
Kubernetes saves Secrets and a lot of other variables as base64 encoded strings. Base64 is just an encoding, and no encryption, meaning we can easily, without any keys, "translate" the above string into the true form. Try it out yourself with tools like e.g. https://www.base64encode.org/!
This means, that if someone gets access to your Kubernetes cluster, he can just read all your secrets.
This is currently not completely preventable, but good solutions exist.
For example "Sealed Secrets", allow you to encrypt your secrets at rest and in transit on creation. The decryption key is stored in the cluster itself, which is where the secrets get decrypted. This still means though, that if the cluster itself gets compromised, the encryption key is compromised, but at least during creation and when storing the secrets on Github, they remain safe.
A slightly more complex approach by Bitnami, the "Helm secrets", can use a Key Management system like AWS KMS or Google CKM to encrypt and decrypt secrets. That way, you do not have to share the same encryption key with all your developers, but only specific developers have access to the encryption keys of their applications, even in the cluster.
Still, the secrets remain "pure" in the cluster itself, which is another reason to pay close attention to not losing your Admin Kubeconfig.

KUBERNETES

SECURITY MISCONCEPTIONS AND MYTHS

5. USING THE SAME USER FOR EVERYTHING

This brings us to our next point. When creating new clusters your provider usually gives you one Kubeconfig file. This is the admin file, and you can do anything with it. Even though it is tempting, you should never use it except for creating other users and permissions. That way, if one user and role for a specific app or namespace gets compromised, it has no control over the others. The solution is called RBAC (role based access control).

6. PERSISTENT VOLUMES WILL AUTOMATICALLY "JUMP" INTO OTHER AVAILABILITY ZONES

Even though Kubernetes is amazing for handling persistence, it must be configured first. Something that could happen when volumes are not properly configured is, that the application will keep running on one node for months, but then if the node goes offline and the deployment automatically switches to another node, it will "start fresh" with the data. This happens, if the persistent volumes are configured to use local storage like many K8s distributions do by default.

To achieve true resistance, one needs to use external volumes like AWS EBS, or block storage in general. You can build your solution on bare-metal servers as well, using tools like Longhorn, but it can pay off to just use the EBS volumes provided, as they are oftentimes backed up to at least three data centers and more.
Luckily, many cloud providers provide the matching settings to be easily applied with kubectl apply, helm, or even include them by default.

Still, try to make sure you are using proper volumes and not just local storage, as this can give you quite a headache some weeks later.

CONTACT

☎ +1 (650) 382 0775

☎ +49 160 1136770

✉ easycloudhost@datafortress.cloud

🌐 easycloudhost.de/services/easykube/

EASY Kube
Managed Kubernetes

KUBERNETES

SECURITY MISCONCEPTIONS AND MYTHS

7. IGNORING RESOURCE LIMITS

I know, it is tempting to just write your deployments and publish them. But you should at least set namespace limits, or even better deployment resource limits, as this prevents you from a lot of headaches further down the road.
Kubernetes managed resources automatically, which is amazing, but if someone attacks one specific service of yours with a DDOS attack (basically a lot of traffic), the resources for that deployment will consume almost all Kubernetes resources and all the other applications will fail.

This can be troublesome if your organization uses one huge cluster without these limits. It does not even have to be an attack. It can be an application with bugs, that enters an infinite loop or just eats all the resources. The solution is to define deployment - or at least - namespace limits. That way, if a namespace reaches that hard limit, the application will be contained, and the other applications keep on running.

CHECKLIST ON THE NEXT PAGE

I hope this list gave you a nice picture of what you can do to improve security in your Kubernetes deployment. What was new to you? Did I miss something? Let me know, and feel free to reach out to me to discuss your Kubernetes projects. Did you know that I am offering a worry-free managed Kubernetes called "EasyKube"? Check it out on https://easycloudhost.de

CONTACT

+1 (650) 382 0775

+49 160 1136770

easycloudhost@datafortress.cloud

easycloudhost.de/services/easykube/

EASY Kube
Managed Kubernetes

KUBERNETES

SECURITY MISCONCEPTIONS AND MYTHS

CHECKLIST

- [ ] Install a CNI plugin to isolate networks
  Linkerd, Calico, Flannel, Weave, or Cilium
- [ ] Save Backups in safe locations, or encrypt them
  If you store sensitive data you can't use AWS, Azure or Google!
- [ ] Encrypt Kubernetes secrets
  Sealed Secrets / Helm Secrets
- [ ] Create a user for each Deployment & Developer
  RBAC
- [ ] Set up persistent volumes to use "real" volumes
  CSI-drivers
- [ ] Set Deployment and Namespace limits
  K8s Resource limits

CONTACT

☎ +1 (650) 382 0775

☎ +49 160 1136770

✉ easycloudhost@datafortress.cloud

🌐 easycloudhost.de/services/easykube/