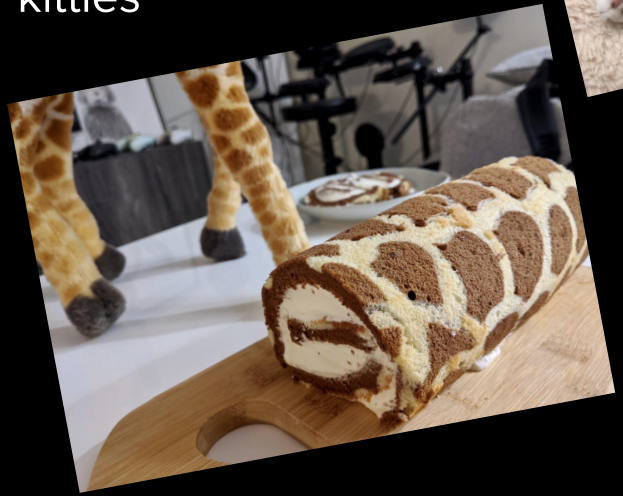
DEVNEXUS™

Bullet-proof Microservices with Kubernetes & Spring

Feb 17 2021
https://devnexus.com/

About us

- Bella Bai
  - Yuxin Bai
  - 白玉欣
- Servant of two rescued kitties

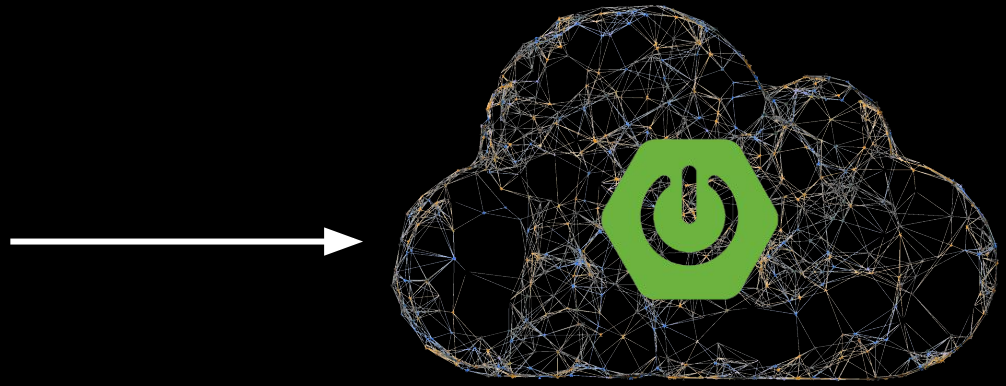


About us

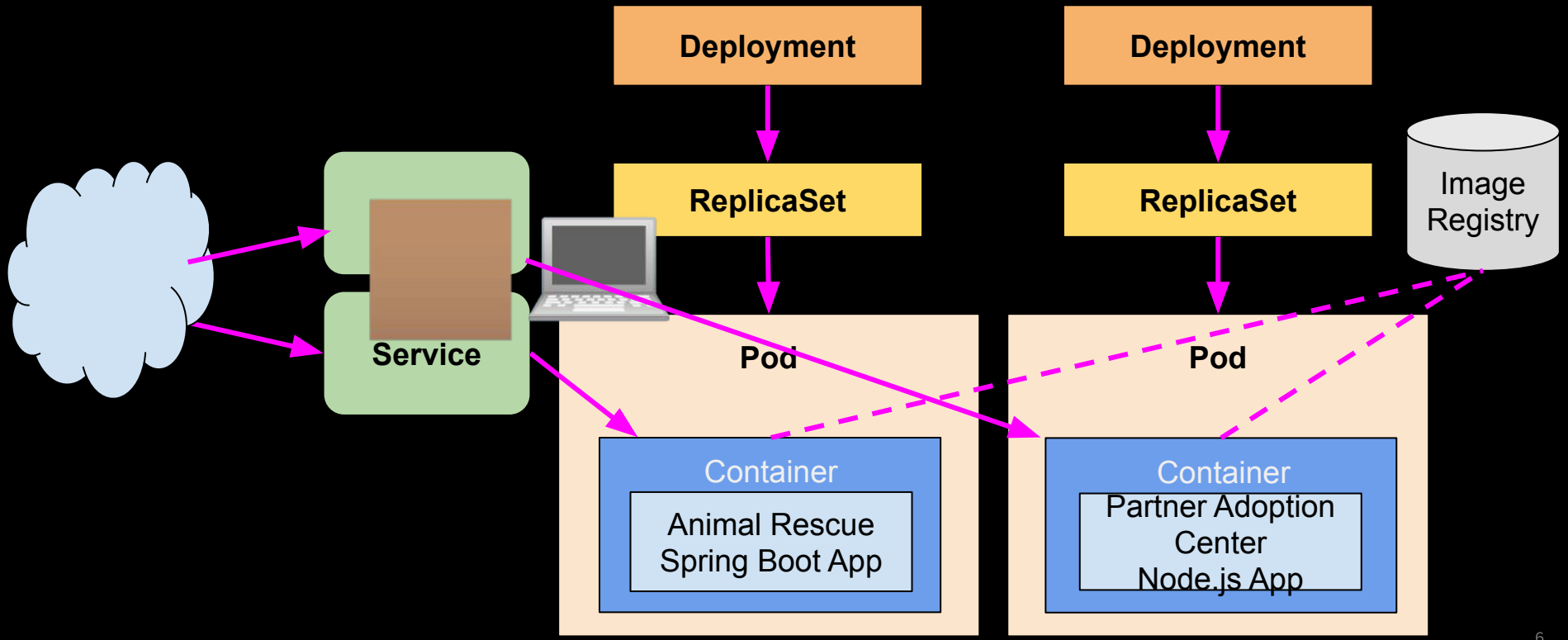
- Ollie Hughes
  - @olliehughes82

What is a Bullet-proof microservice?

As developers, we want... 💻 🍕 🎮



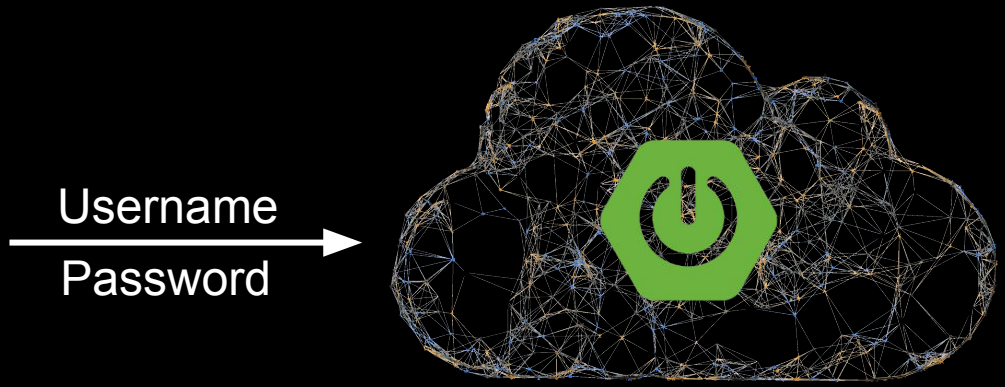
Deploying apps to Kubernetes



Demo - Deploy Animal Rescue

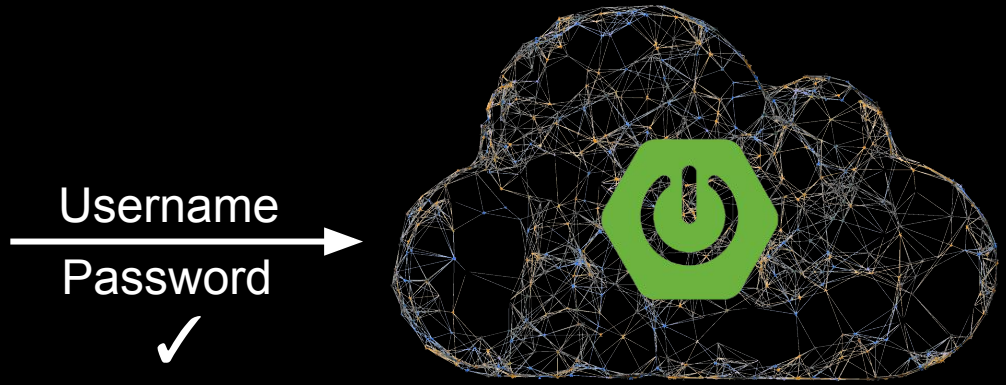
https://github.com/LittleBaiBai/animal-rescue

Basic Authentication

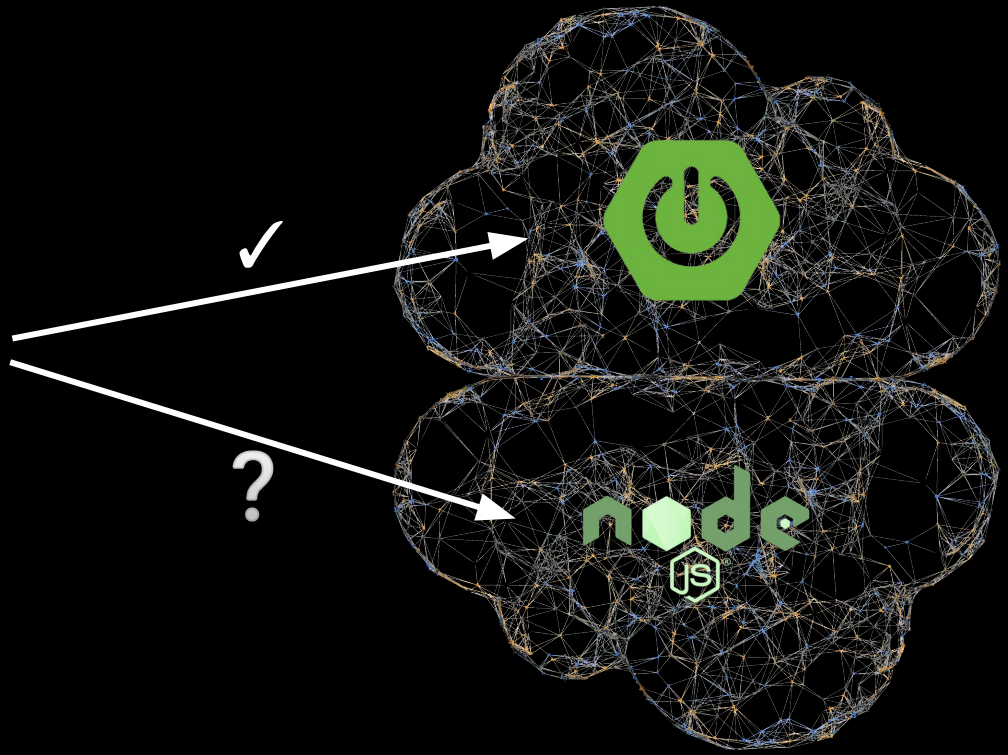


Demo - Add Basic Auth

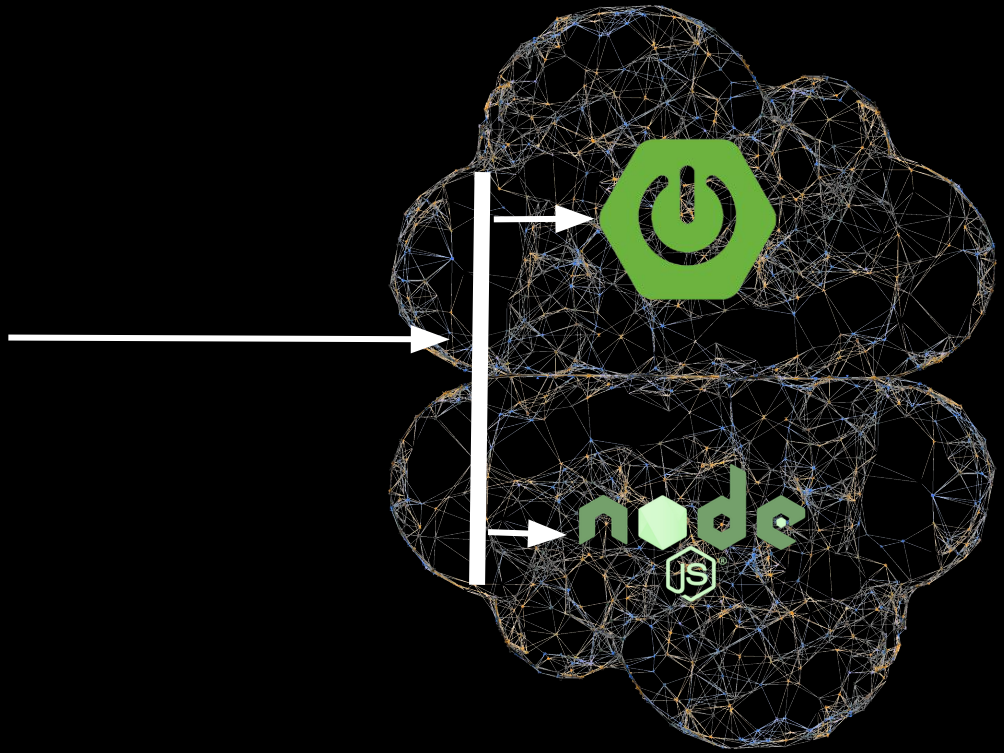
So, back to this diagram...



Non-Spring apps?

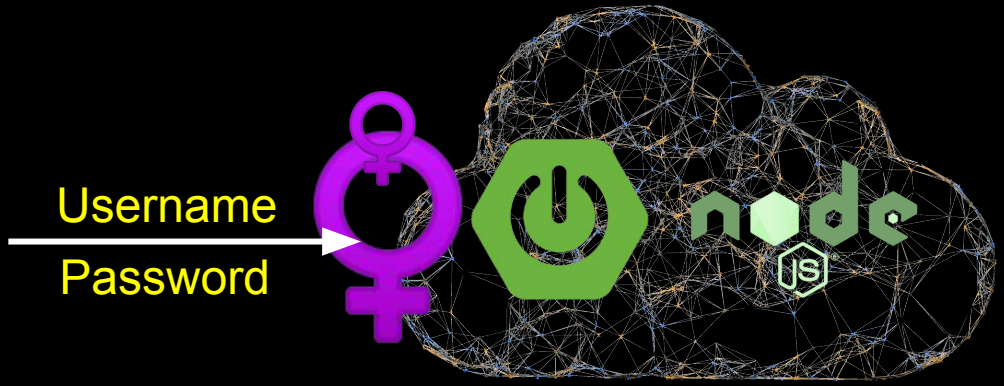


Add a layer in between!



Demo - Use Ingress

Wait, HTTP is not encrypted...



Need to enable TLS, but how?

- Pick a Certificate Authority (CA)

- Manage certificates (issue/update/renew)

- Enable TLS on servers

Or

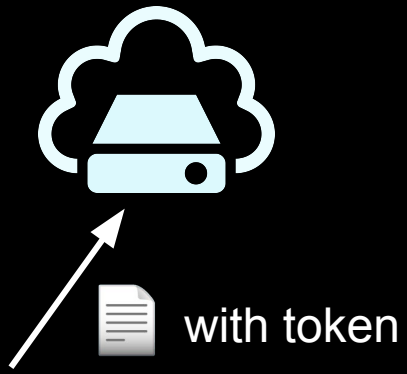
Ingress

ACME HTTP01 Challenge

Need a cert for animalrescue.online plzzzz.

ACME HTTP01 Challenge

ACME HTTP01 Challenge



ACME HTTP01 Challenge

Ready for action!

ACME HTTP01 Challenge

Retrieve and verify the file from http://animalrescue.online/.well-known/acme-challenge/<TOKEN>

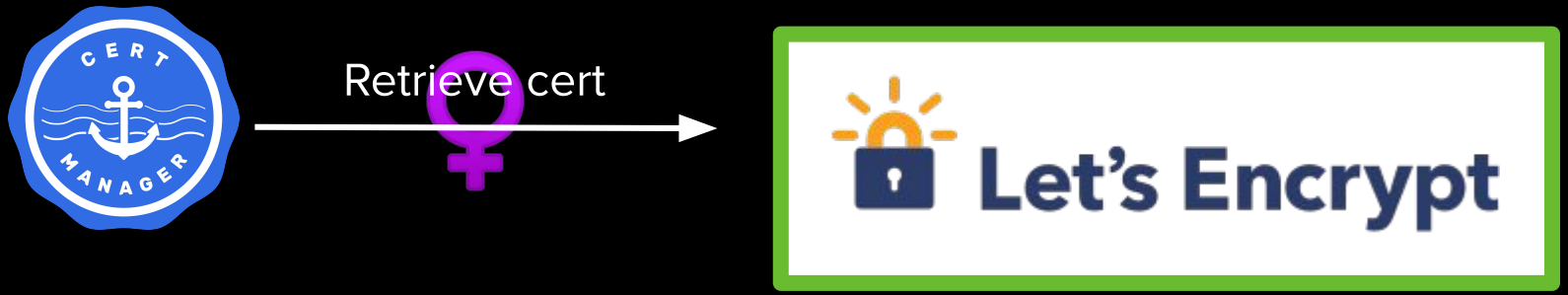
ACME HTTP01 Challenge

I trust this account.
I will grant a valid cert -
On their next request.

ACME HTTP01 Challenge

Retrieve cert

ACME HTTP01 Challenge

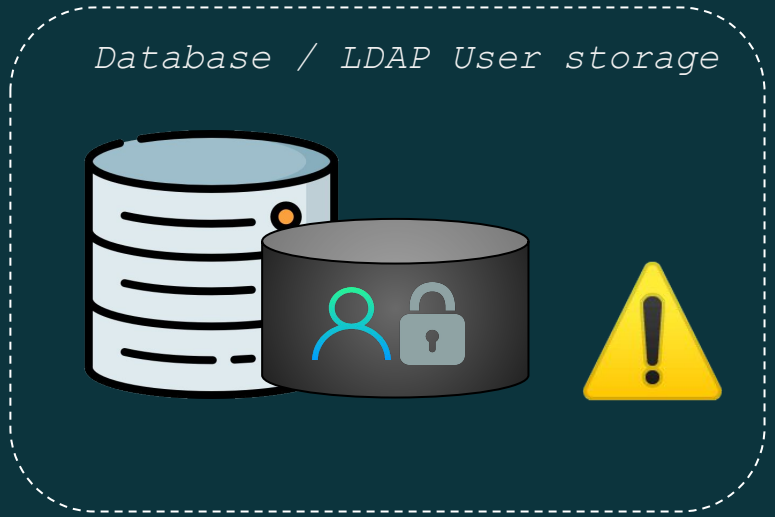


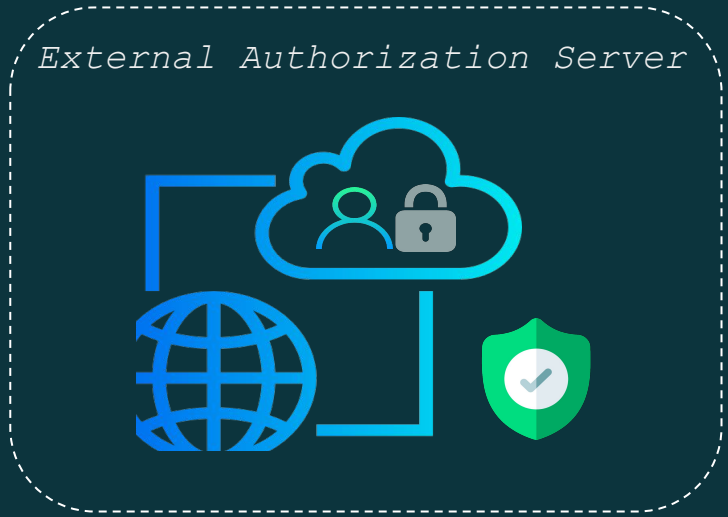
Demo - Enable TLS

Now the communication is secured

But I want to use my GitHub account!

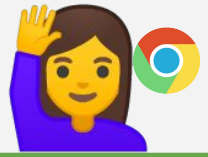
OAuth 2





External Auth Is Better Than Local

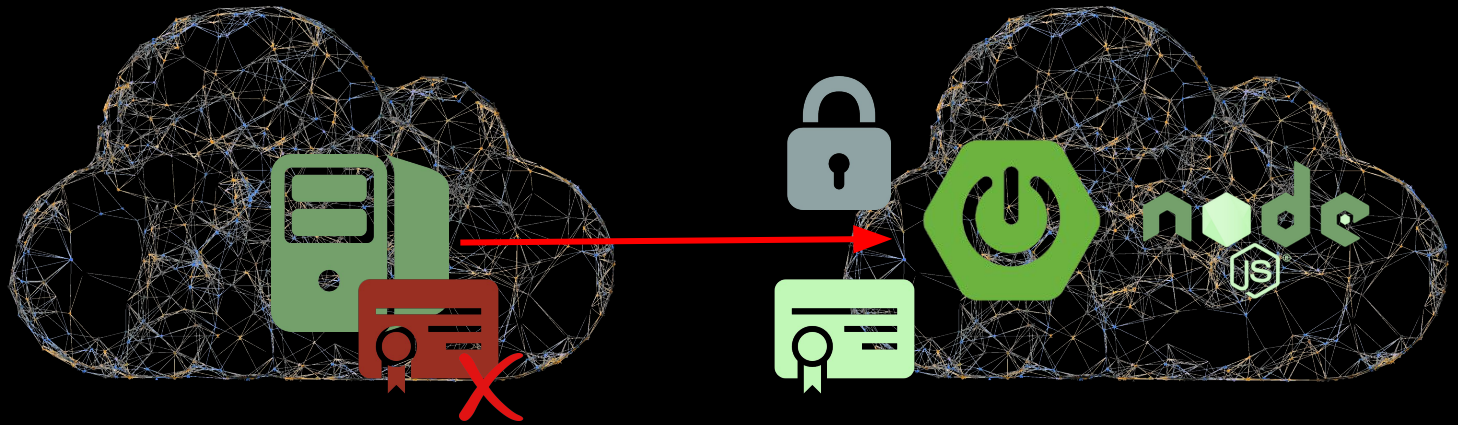
OAuth 2 Login Flow



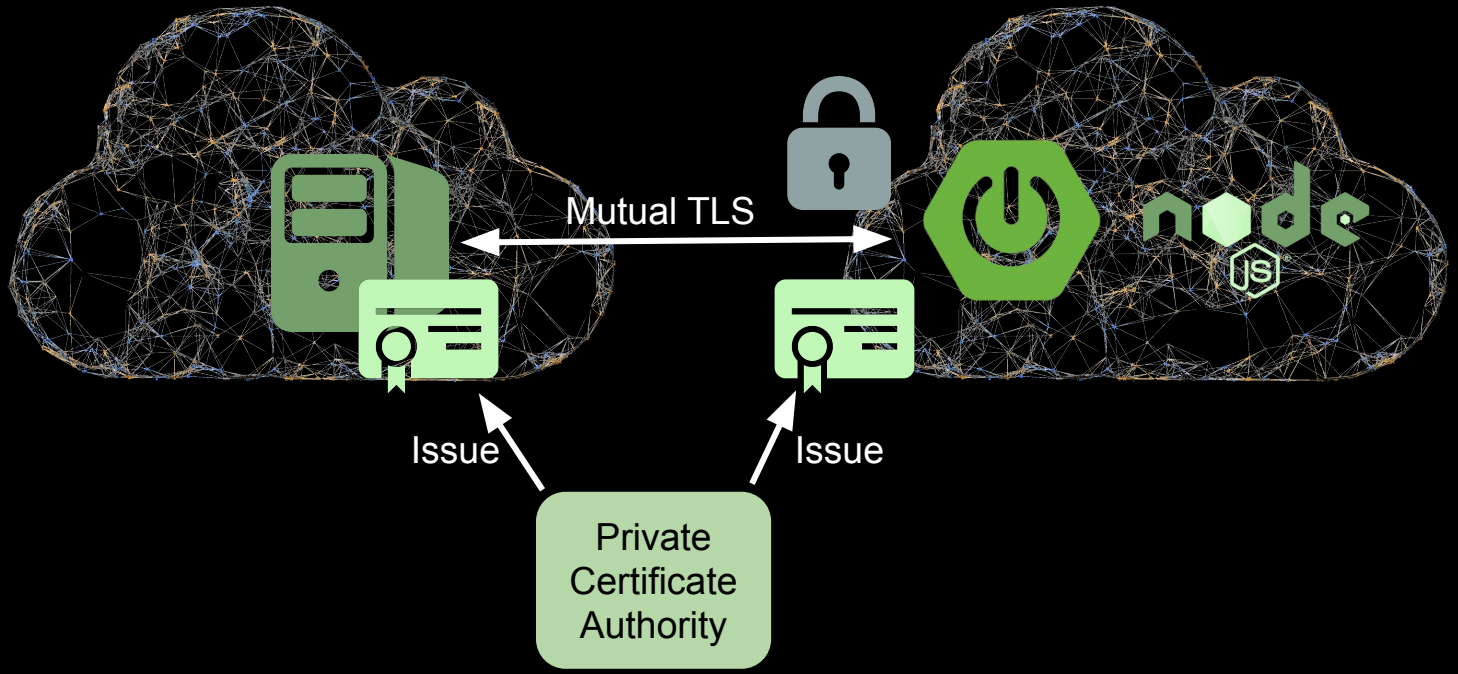
User	Spring Cloud Gateway (Resource Server)	Auth0 (Authorization Server)	Animal Rescue App
1. User browses to route `https://gateway...`	2. Gateway sends 302 redirect to the registered client sign in page on Auth0		
3. User enters Auth0 credentials		4. Redirect to Gateway sent containing authorization code	
	5. Authenticate	6. Issue access token	
			7. Request is authorized, display username from embedded token
8. User can access the resource ✔️			

Demo - OAuth 2
with Spring Cloud Gateway

But OAuth2 is for users, what about machines?



Clients can prove their identities too!



How to mTLS in k8s?

Create CA certificate
with a self-signed issuer

Create CA issuer
with the created CA certificate

Issue certificates
with the created CA issuer

Mount certificates
on app deployments

Update apps
to use the mounted certificate for mTLS

Step by step guide: https://blog.jetstack.io/blog/securing-mysql-with-cert-manager/

Autocert makes mTLS easier

Create CA certificate
with a self-signed issuer

Create CA issuer
with the created CA certificate

Create certificates
with the created CA issuer

Mount certificates
on app deployments

Update apps
to use the mounted certificate for mTLS

Demo - mTLS with Autocert

A few notes about Autocert

- Pros:
  - Certificates are generated and only available within the pod
  - Easy to set up and run - just one annotation needed
  - Perfect for apps that already know how to handle mTLS

- Cons:
  - Need to add some code to:
    - Load the certs and keys
    - Watch for file changes on cert rotation
  - No fine-grained access control

Automating mTLS with Service Mesh

Tanzu Service Mesh

Tanzu Service Mesh: https://docs.vmware.com/en/VMware-Tanzu-Service-Mesh/services/concepts-guide/GUID-9E3F1F90-4310-415B-98C8-C06E59B8A5EE.html
Traefik: https://docs.traefik.io/https/tls/#client-authentication-mtls
Istio: https://istio.io/latest/docs/concepts/security/#mutual-tls-authentication
Linkerd: https://linkerd.io/2/features/automatic-mtls/

Check out our repo

https://github.com/LittleBaiBai/animal-rescue

Stay Connected.

Bella Bai

LittleBaiBai bellalleb_bai

Oliver Hughes

@olliehughes82