API MANAGEMENT IN THE CONTEXT OF HYBRID CLOUD

Francesco Semeraro | Cloud Engineer @Cluster Azure IT

Politecnico di Torino
REPLY

WHO AM I?

Francesco Semeraro

Cluster Azure Reply

- Cloud Engineer & Solutions Architect

PoliTO

- 2nd level Master's degree – AI & Cloud

PoliMI

- Master's degree – Computer science eng.

PoliBA

- Bachelor's degree – Computer science eng.

franckies.github.io
f.semeraro@reply.it

OUTLINE

Introduction

- What is an API?
- API Management
- Cloud, On-premises and Hybrid approaches

Hybrid API Management

- Architecture
- Cloud side - Azure API Management
- On-prem side - Self-hosted gateway

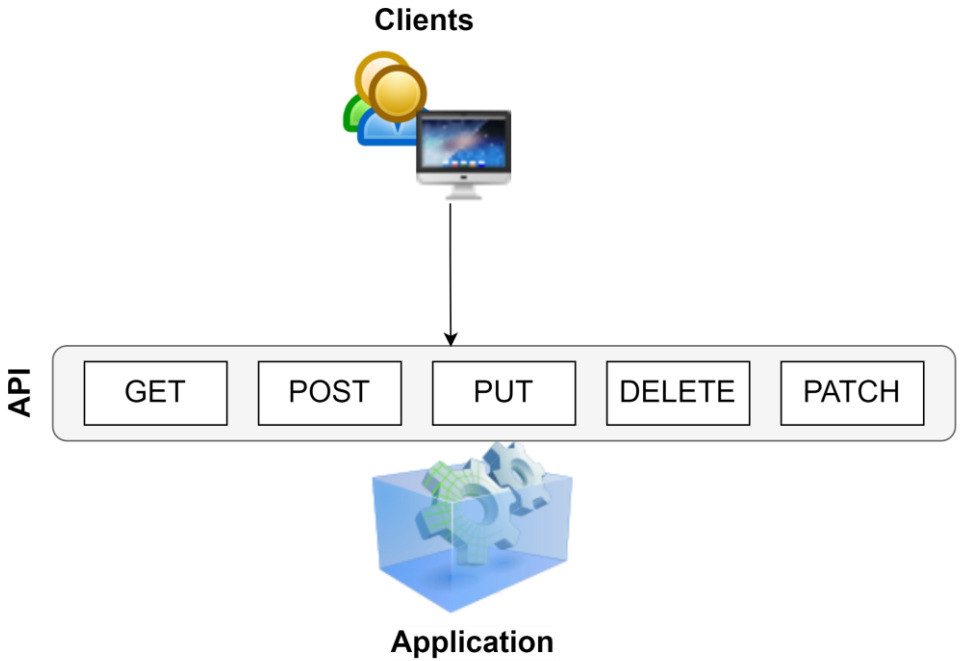
Demo

Conclusions

INTRODUCTION

WHAT IS AN API

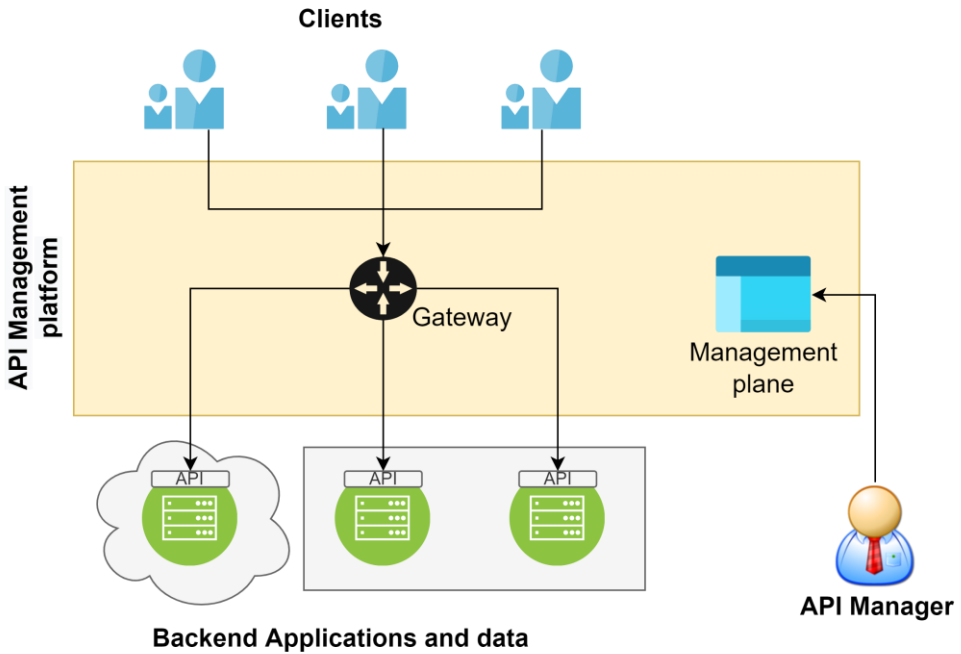
- **Application Programming Interfaces** (APIs): programming interfaces that allow users to access services, which can be simple **data** or **functionality**.
- Layer of **abstraction** between the **client** and the **services**, allowing the application developer (consumer) to disregard the coding behind an API.



API MANAGEMENT

DEFINITION

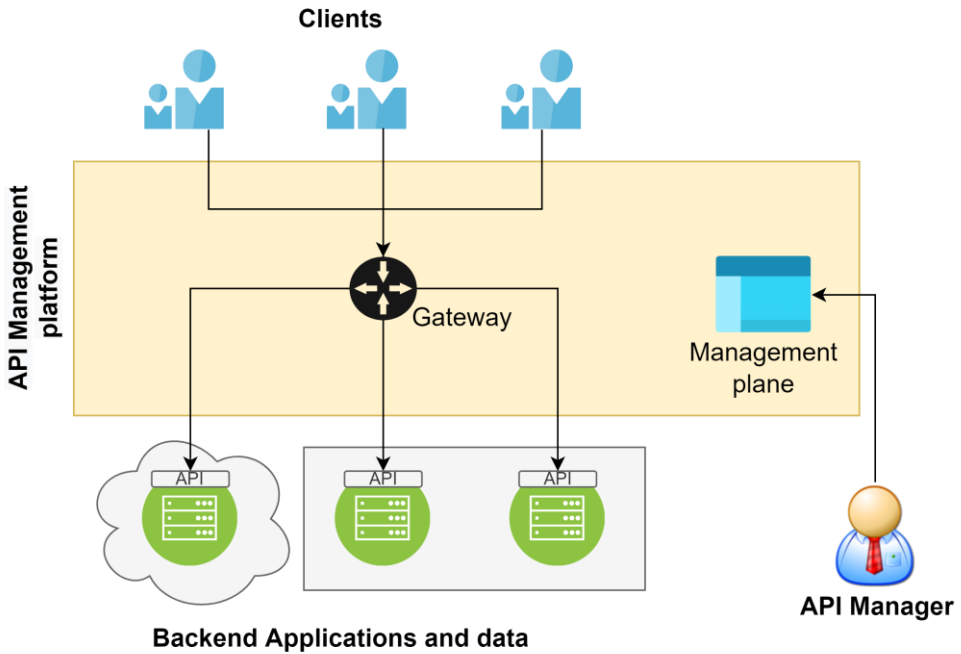
- **API management**: Exposes applications through well-defined **APIs**.
- **Backend apps** and **data** spread across multiple on-premises and cloud environments.
- Consistent and unified management of all API interfaces.



API MANAGEMENT

ADVANTAGES

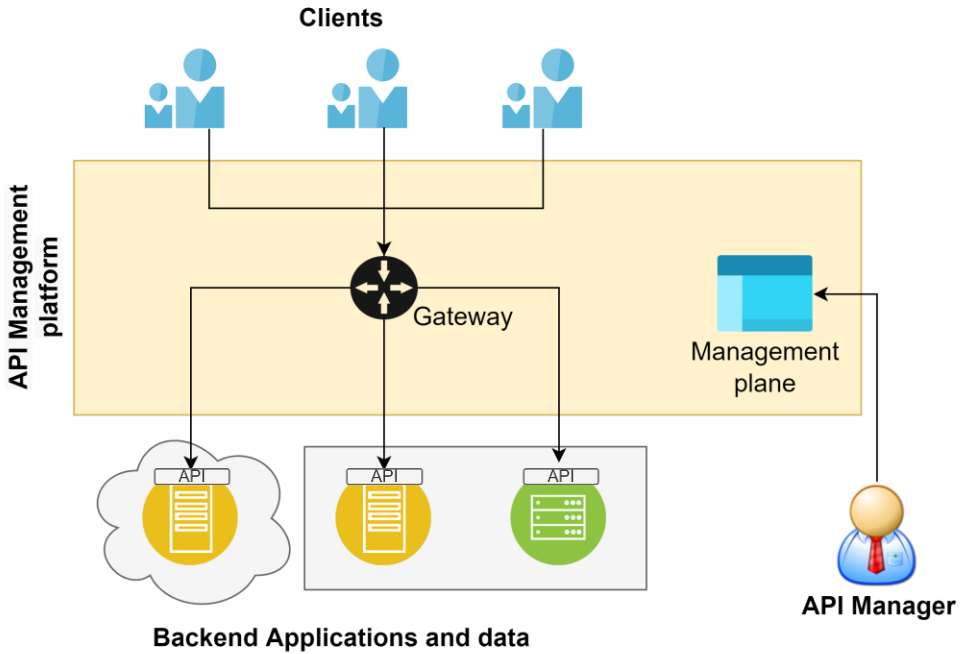
- **Self-service user onboarding:** clients can easily learn about APIs.



API MANAGEMENT

ADVANTAGES

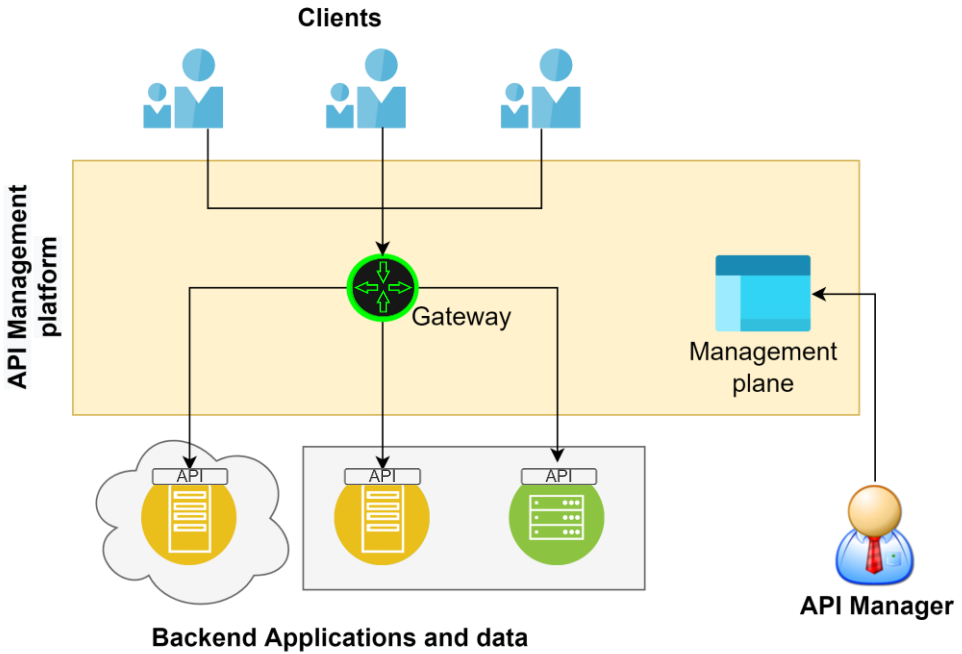
- **Self-service user onboarding:** clients can easily learn about APIs.
- **Façade hiding backends from clients:** you can move or re-architect backends without impacting clients.



API MANAGEMENT

ADVANTAGES

- **Self-service user onboarding:** clients can easily learn about APIs.
- **Façade hiding backends from clients:** you can move or re-architect backends without impacting clients.
- **Single point of entry for backend apps:** this allows to route clients requests in a smart way, providing authN, authZ, and flow control.

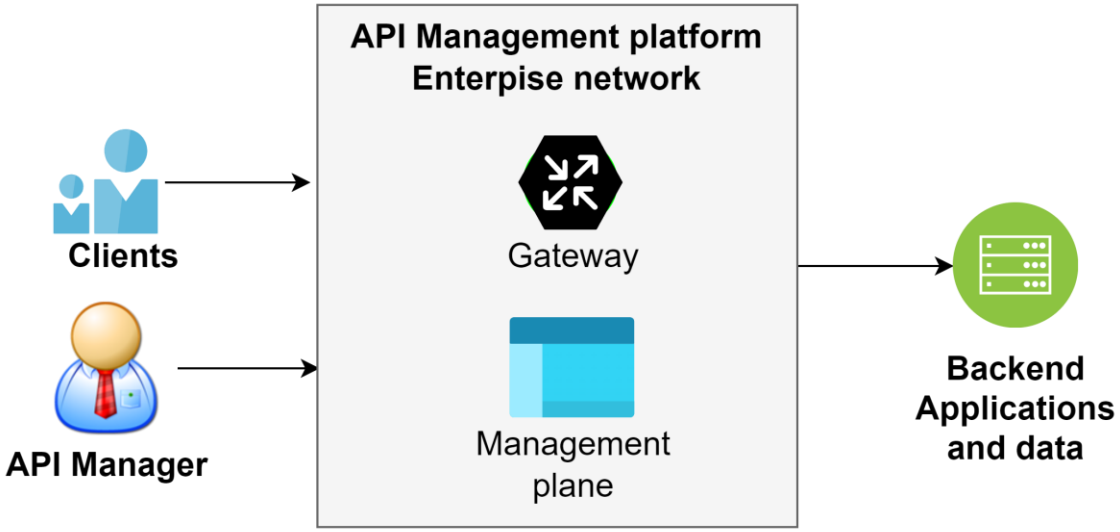


API MANAGEMENT

CLASSIC APPROACHES

On-prem hosted

- Control over the platform
- Reduced latency
- Difficult to maintain
- High TCO

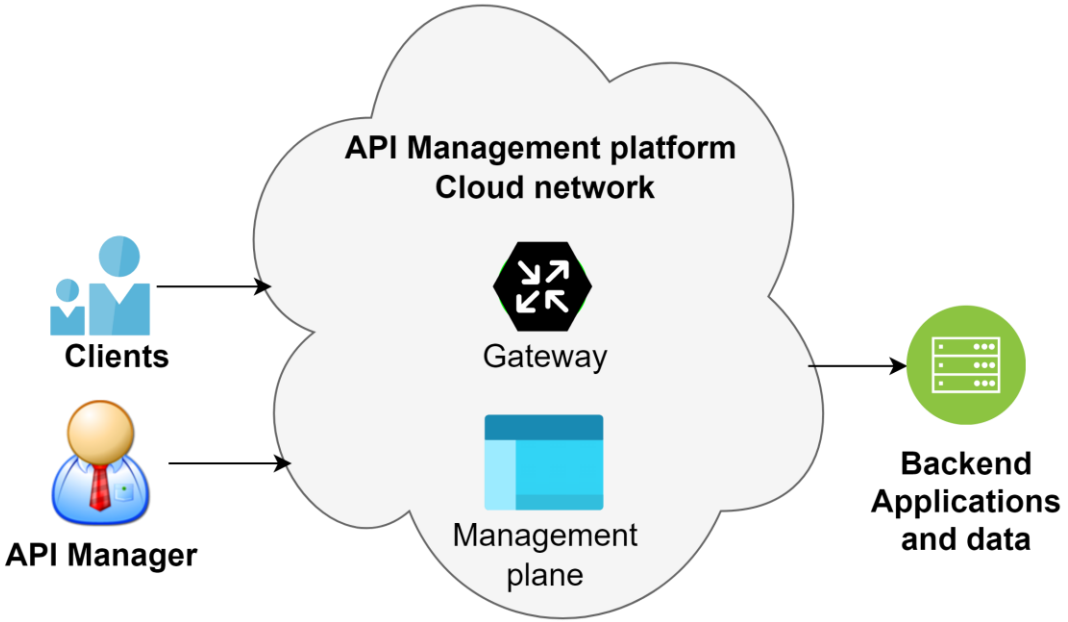


API MANAGEMENT

CLASSIC APPROACHES

Cloud-hosted

- No management costs
- Inefficient for internal APIs
- Compliance and security problems



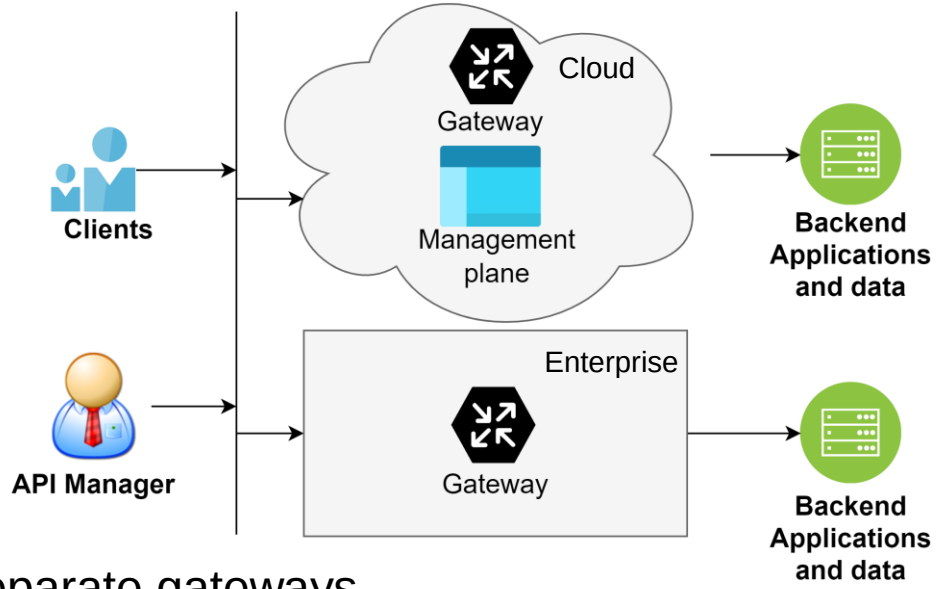
API MANAGEMENT

HYBRID API MANAGEMENT

Hybrid API Management

Advantages:

- No network overhead
- Secure by design
- Cheaper TCO
- Flexible and versatile



Challenges:

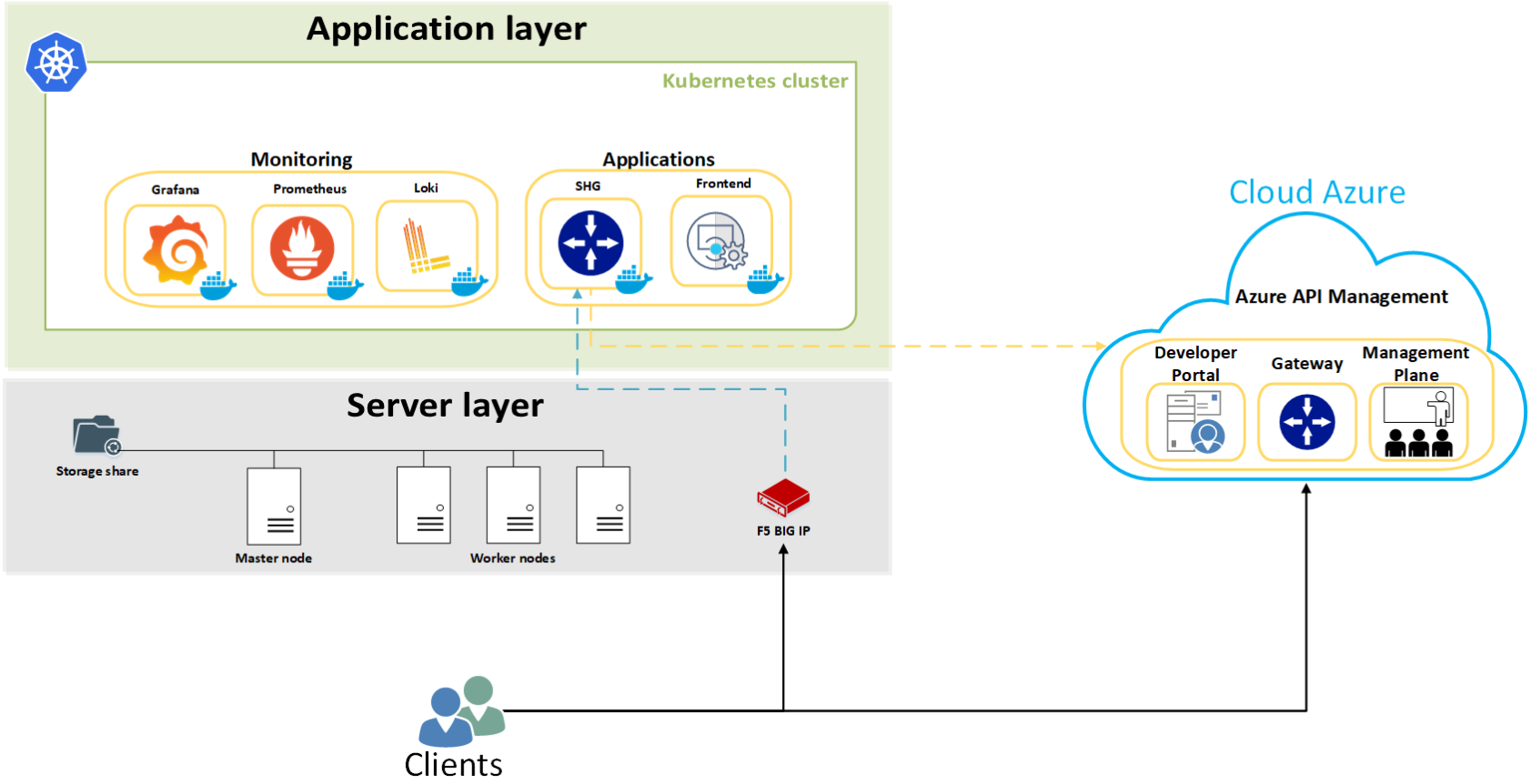
- Configuring and running many separate gateways.
- Avoid on-premises – cloud communication.

➢ It's **crucial to select the right API Management platform**.

HYBRID API MANAGEMENT SOLUTION

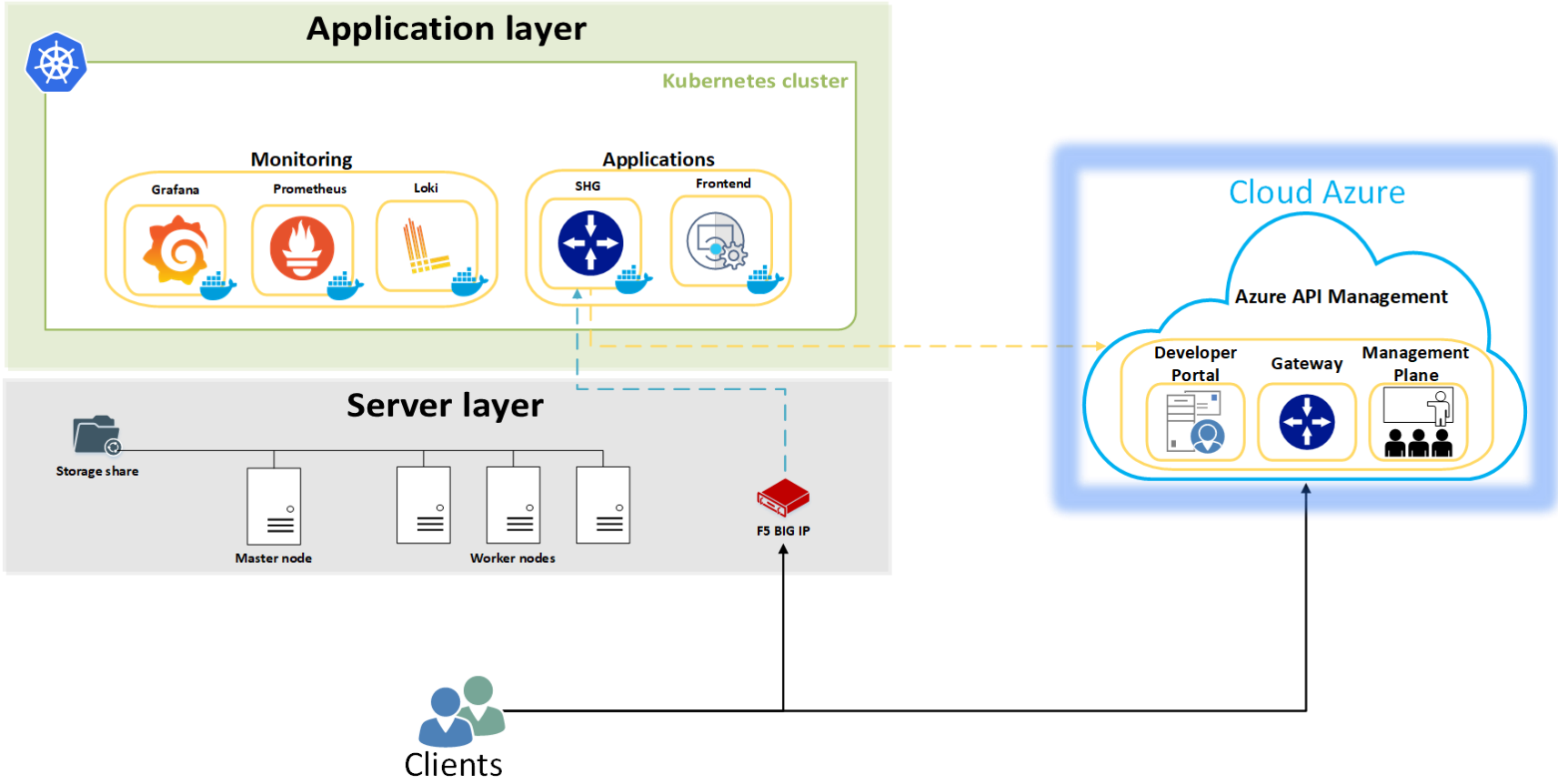
ARCHITECTURE

OVERVIEW



CLOUD SIDE

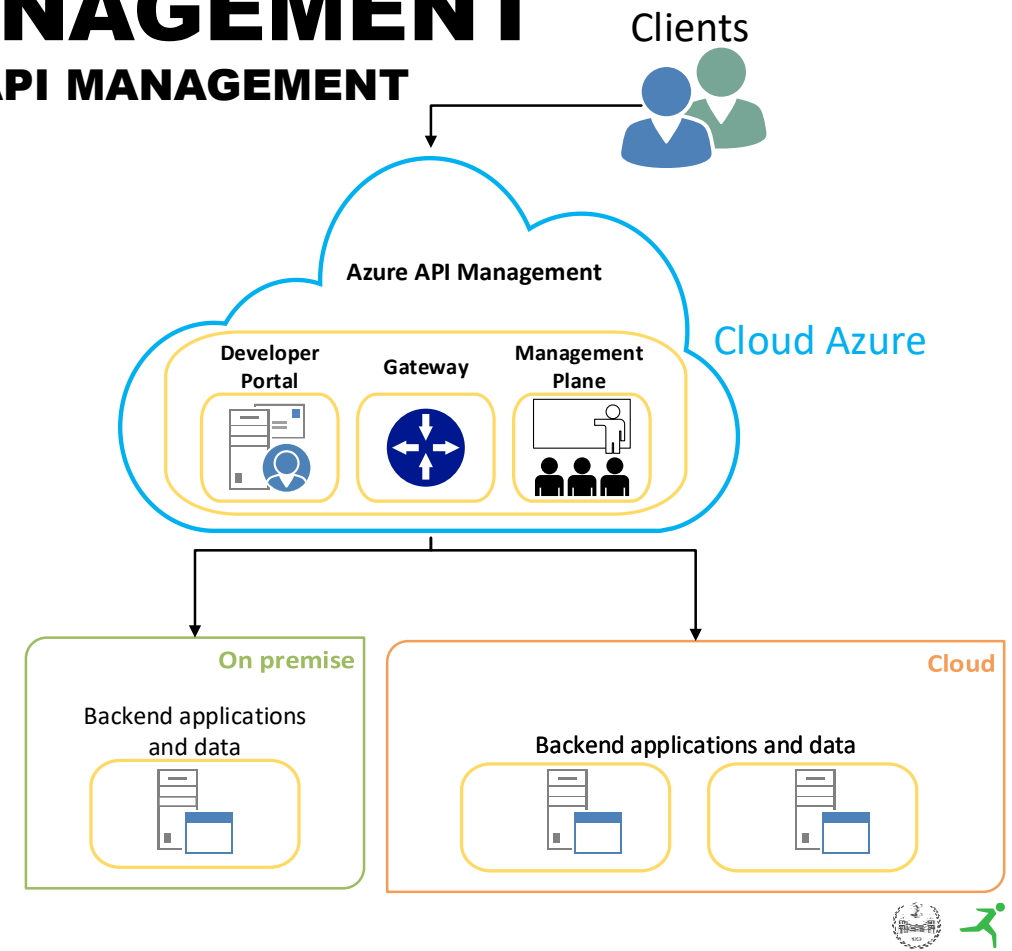
ARCHITECTURE



API MANAGEMENT

AZURE API MANAGEMENT

- Azure API management sits between clients and backend applications and data.



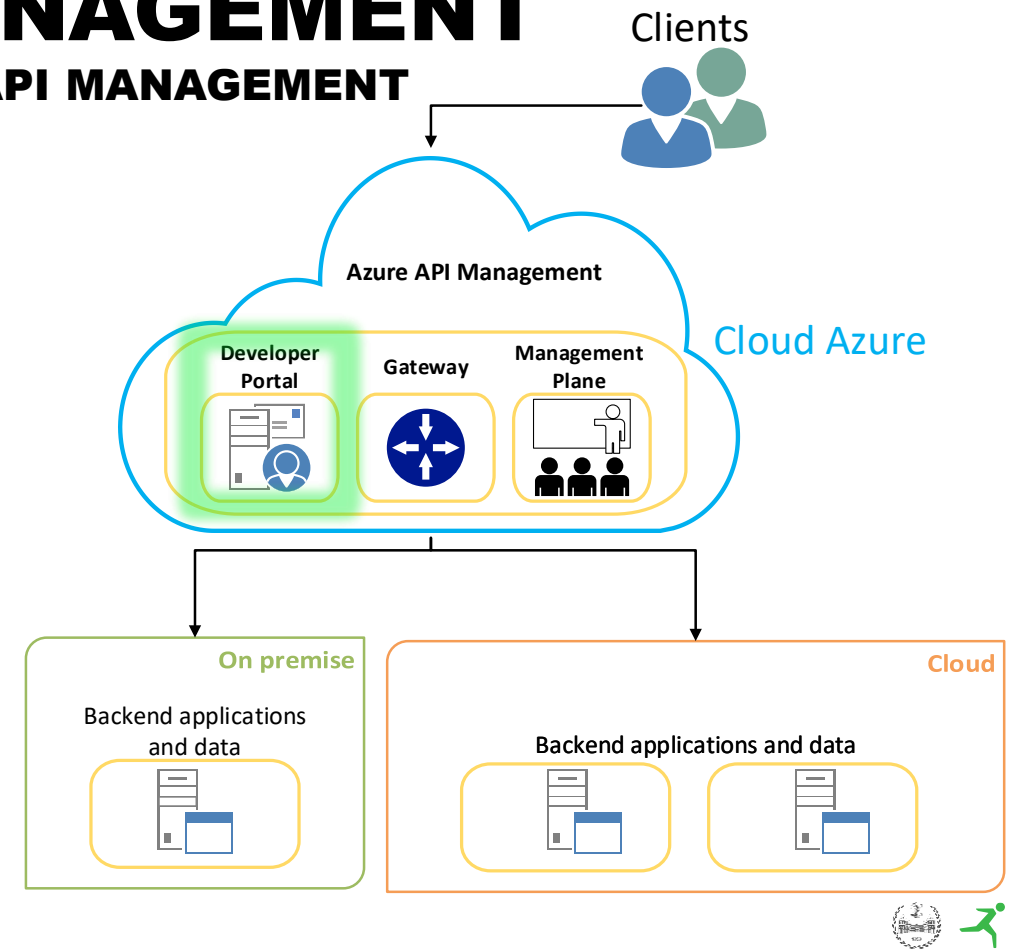
API MANAGEMENT

AZURE API MANAGEMENT

- Azure API management sits between clients and backend applications and data.

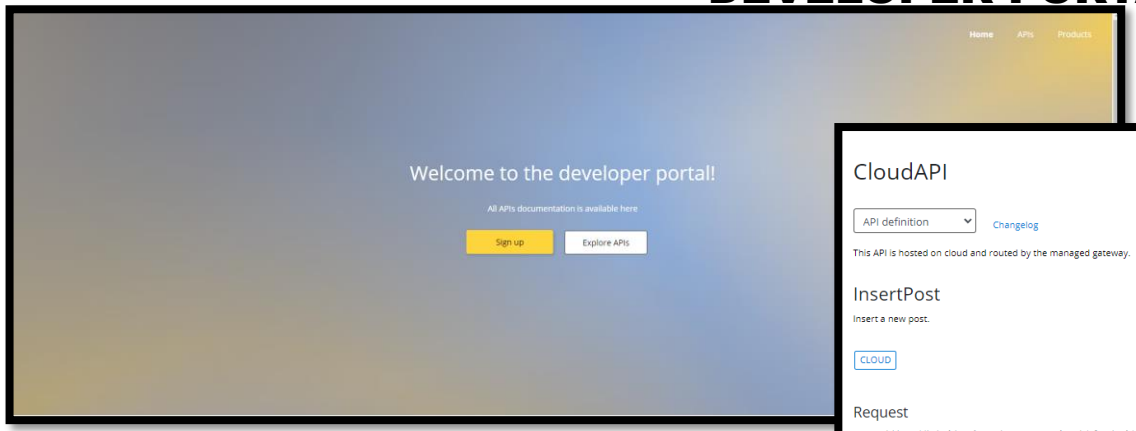
Developer portal

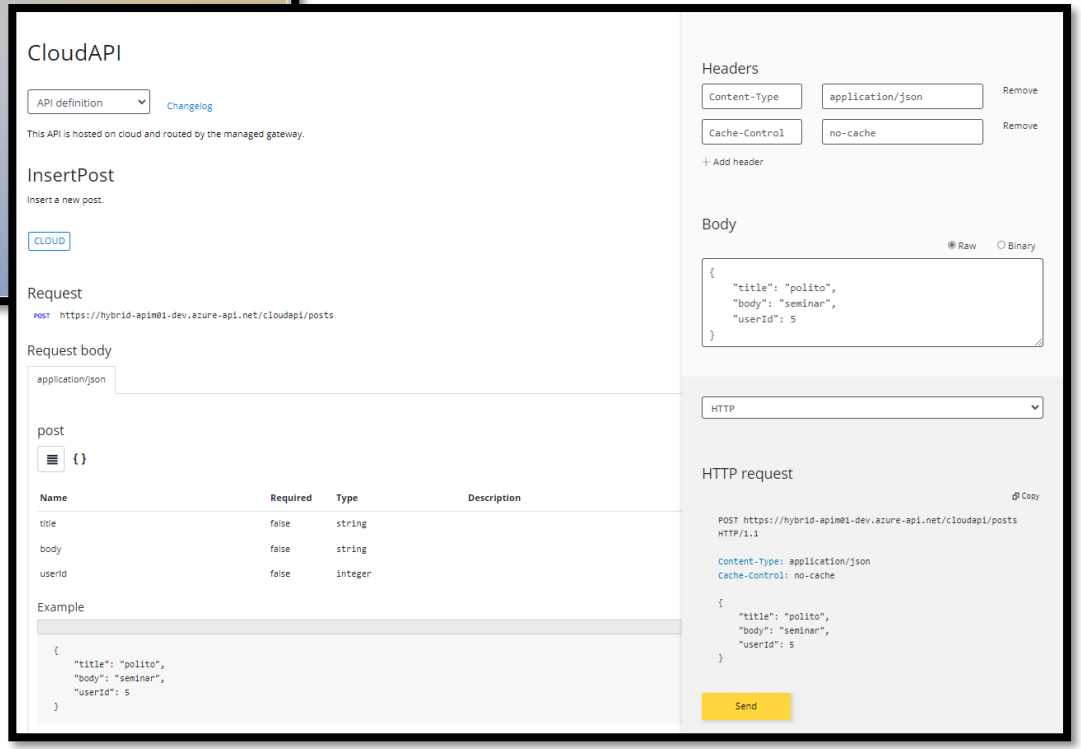
- Contains APIs documentation.



API MANAGEMENT

DEVELOPER PORTAL





API MANAGEMENT

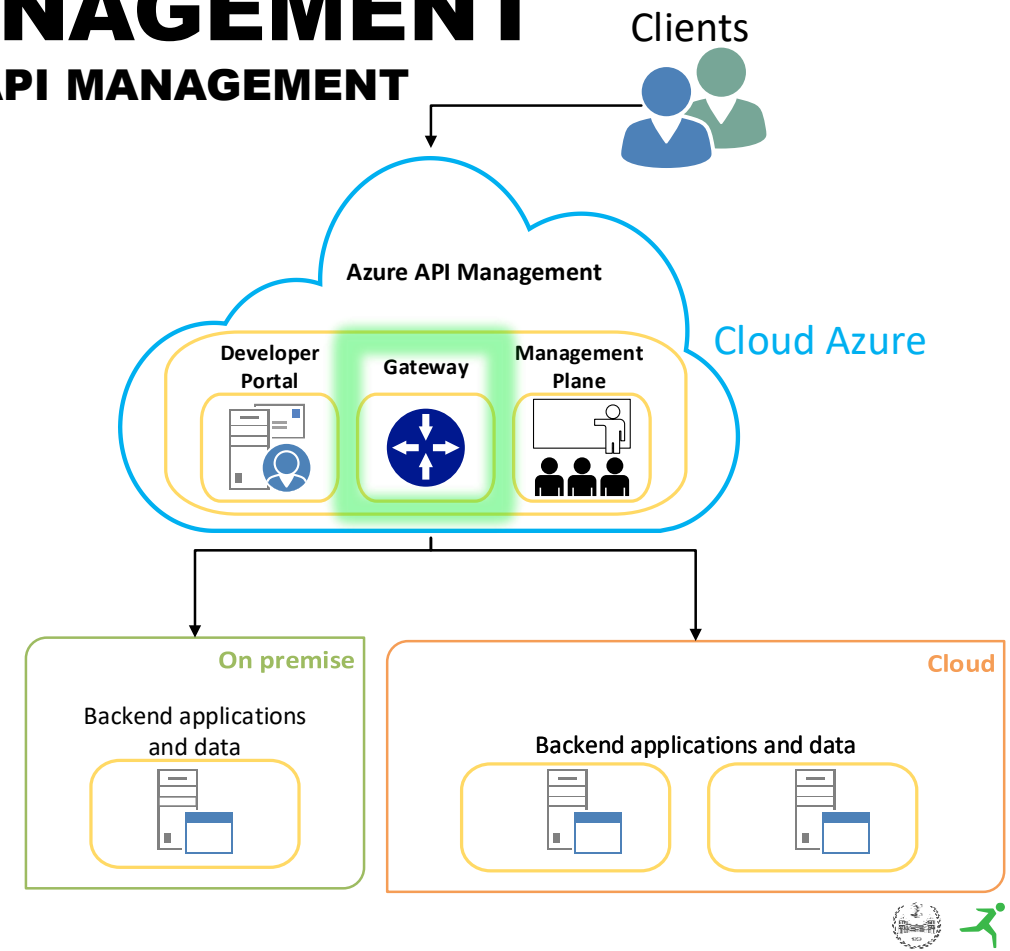
AZURE API MANAGEMENT

- Azure API management sits between clients and backend applications and data.

Developer portal

Gateway

- Provides façade proxies intercepting clients calls and passing them to backend apps.



API MANAGEMENT

AZURE API MANAGEMENT

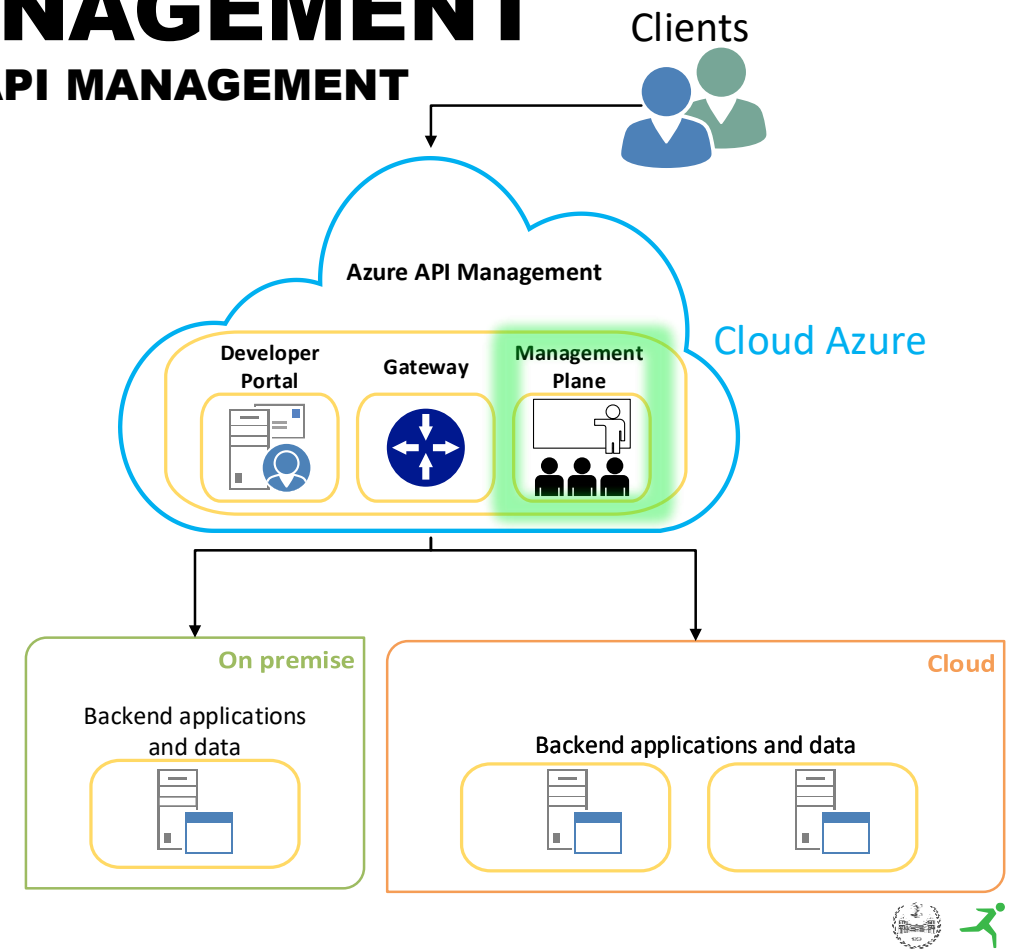
- Azure API management sits between clients and backend applications and data.

Developer portal

Gateway

Management plane

  - Provides analytics that let you monitor the usage and health of your published APIs, together with **control policies**.



API MANAGEMENT

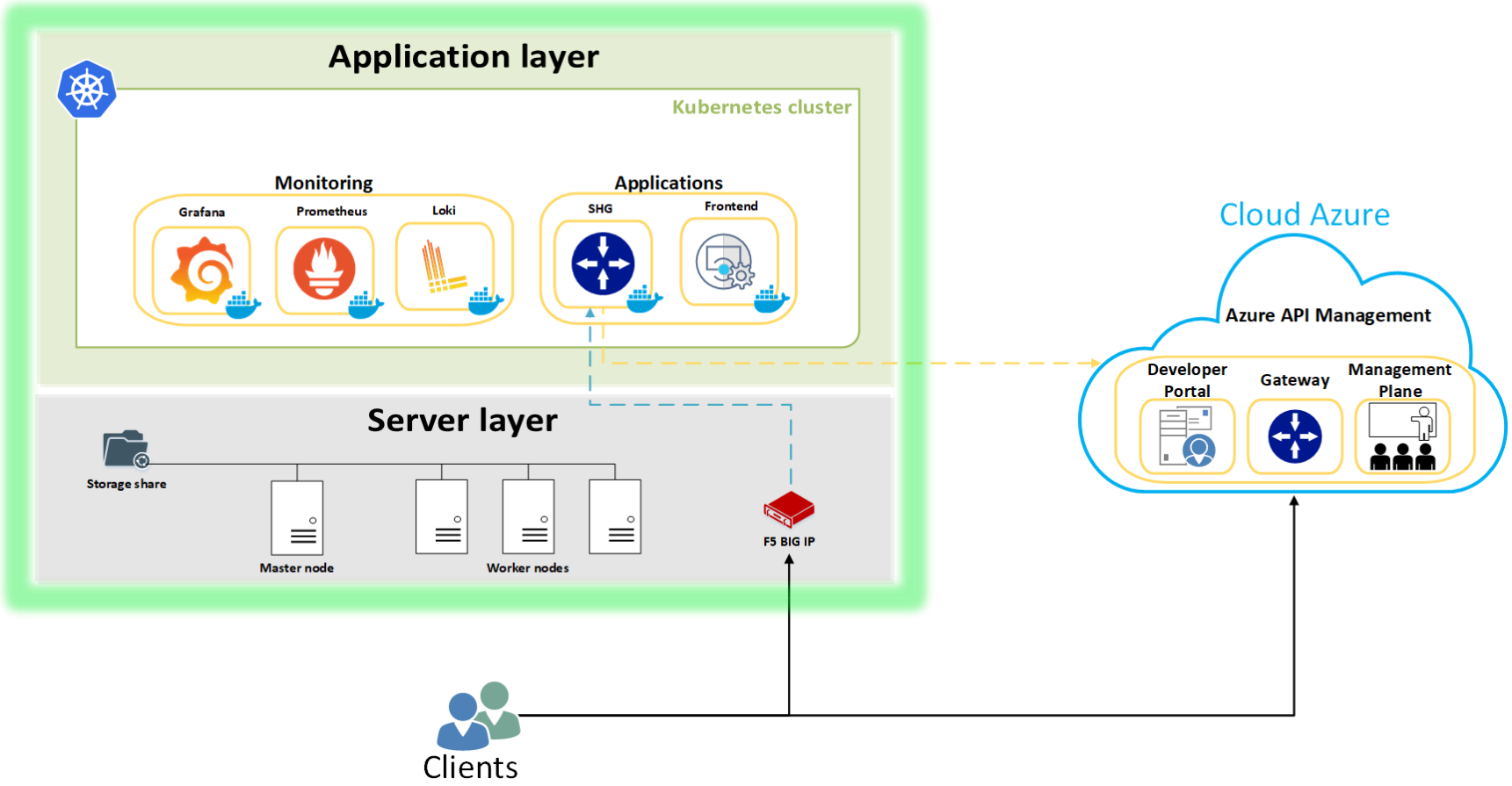
CONTROL POLICIES

```
<policies>
    <inbound>
        <rate-limit-by-key calls="5" renewal-period="1" counter-key="@(context.Request.IpAddress)" />
        <rate-limit-by-key calls="100" renewal-period="60" counter-key="@(context.Request.IpAddress)" />
        <rate-limit-by-key calls="5" renewal-period="1" counter-key="@(context.Subscription?.Key ?? "anonymous")" />
        <rate-limit-by-key calls="100" renewal-period="60" counter-key="@(context.Subscription?.Key ?? "anonymous")" />
        <ip-filter action="allow">
            <address>13.66.201.169</address>
        </ip-filter>
    </inbound>
    <backend>
        <forward-request />
    </backend>
    <outbound />
    <on-error />
</policies>
```

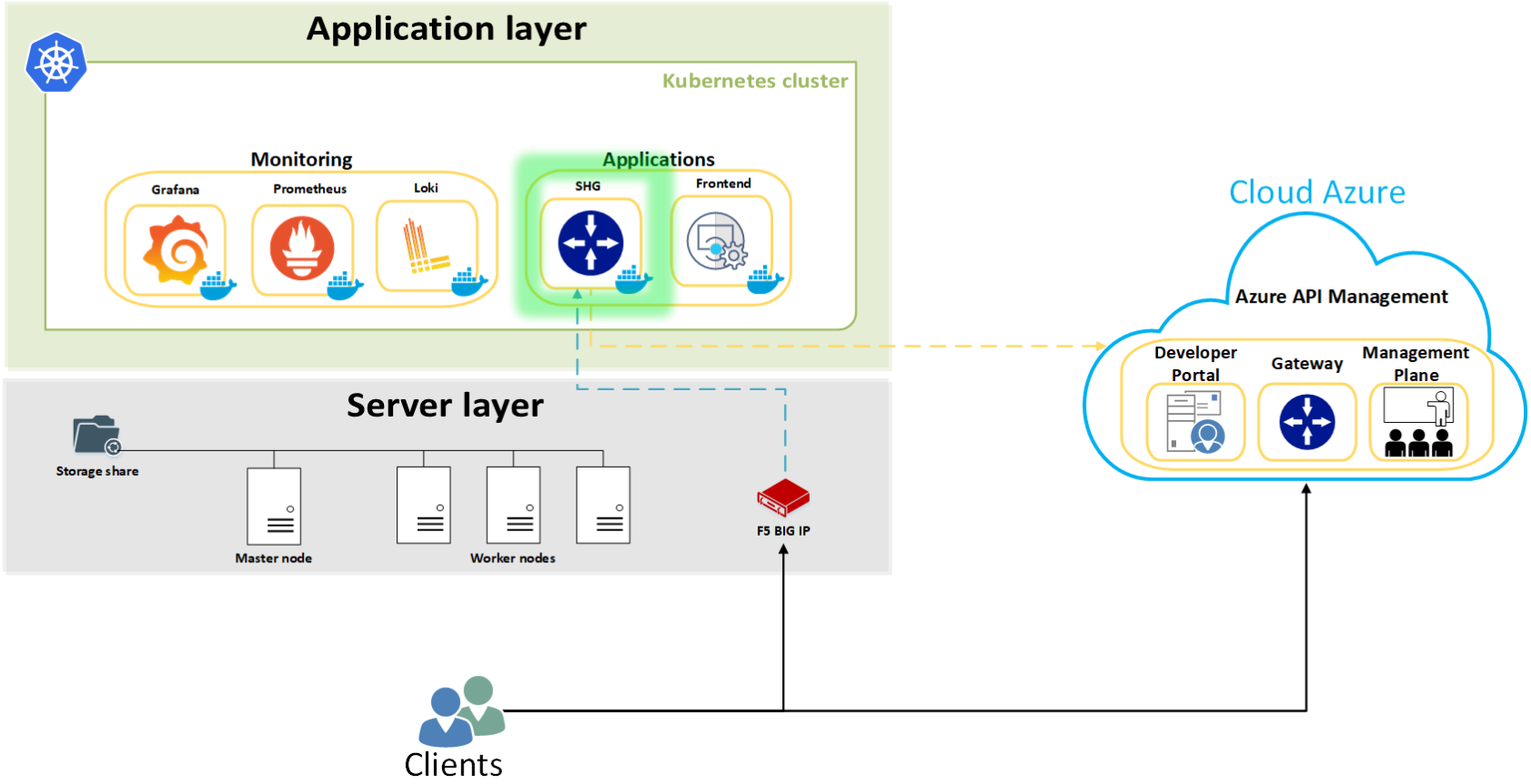
```
-->
<policies>
    <inbound>
        <base />
        <set-backend-service base-url="CHANGEBACKENDURL" />
        <authentication-certificate certificate-id="SSLCERTIFICATEID" />
    </inbound>
    <backend>
```

ON-PREMISES SIDE

ARCHITECTURE



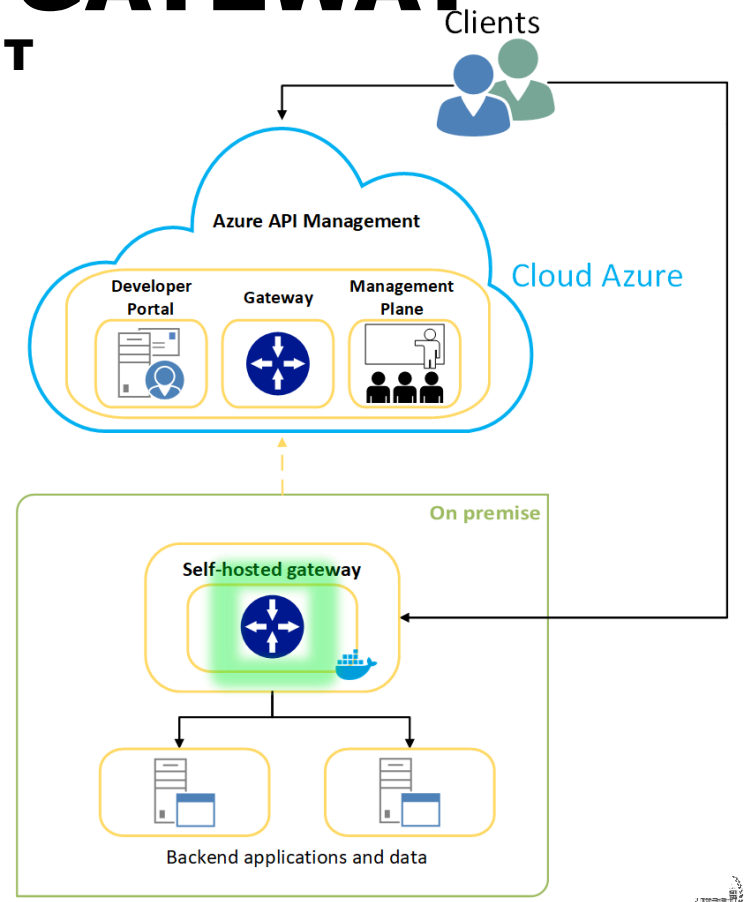
ARCHITECTURE



SELF HOSTED GATEWAY

WHAT IS IT

- On-premise Microsoft solution to enable **Hybrid API Management**.
- It includes only the gateway functionalities**.**
- Developer portal and Management plane shared with the Azure API Management.

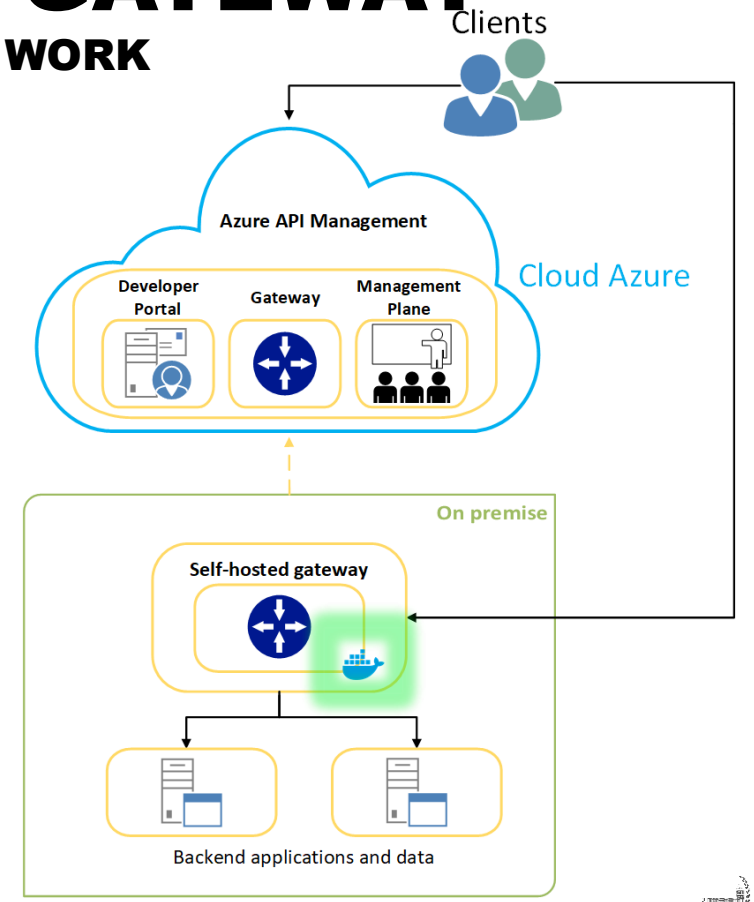


SELF HOSTED GATEWAY

HOW DOES IT WORK

Containerized approach

- Microsoft packaged gateway functionalities in a Linux-based Docker image. Deployed in high availability with Kubernetes to handle scaling, upgrades and more.



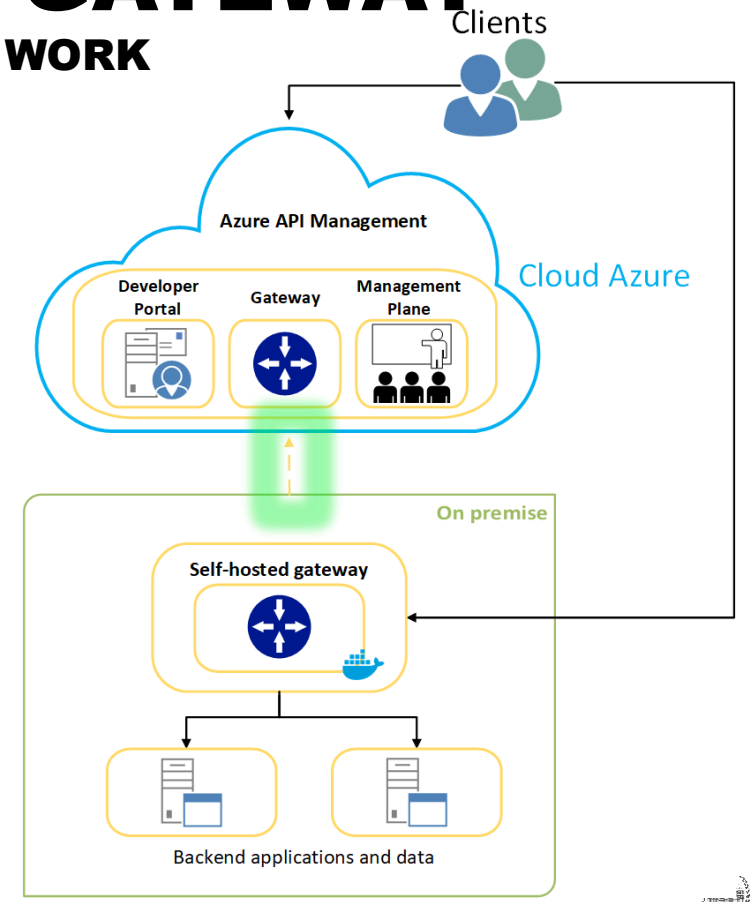
SELF HOSTED GATEWAY

HOW DOES IT WORK

Containerized approach

Only outbound connection

- Avoids the potential security risk of opening an inbound connection while still allowing two-way interaction between the self hosted gateway and the Azure cloud.



SELF HOSTED GATEWAY

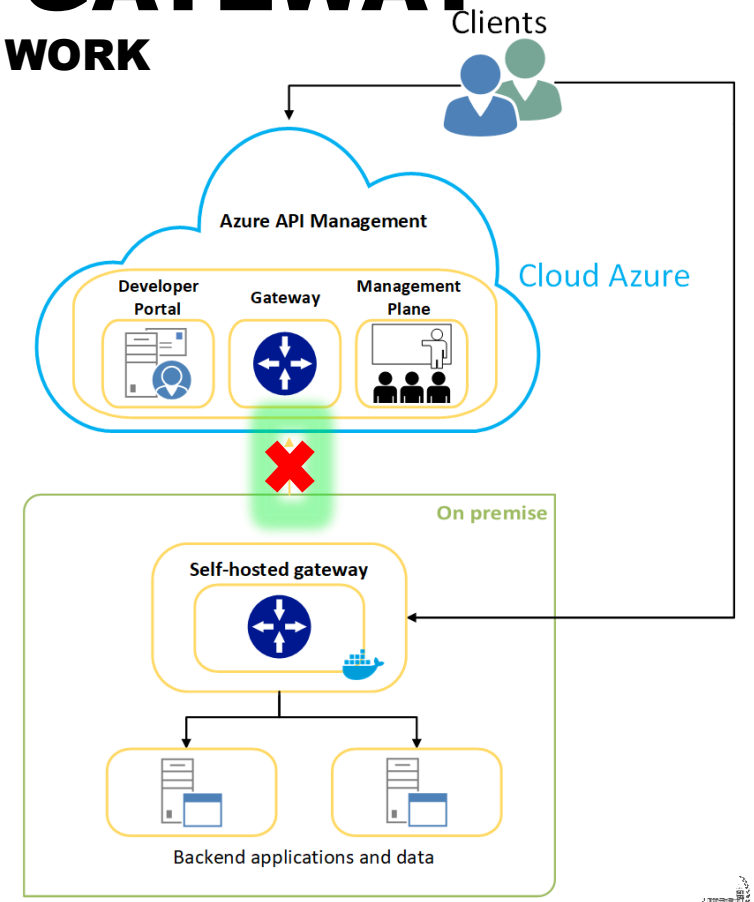
HOW DOES IT WORK

Containerized approach

Only outbound connection

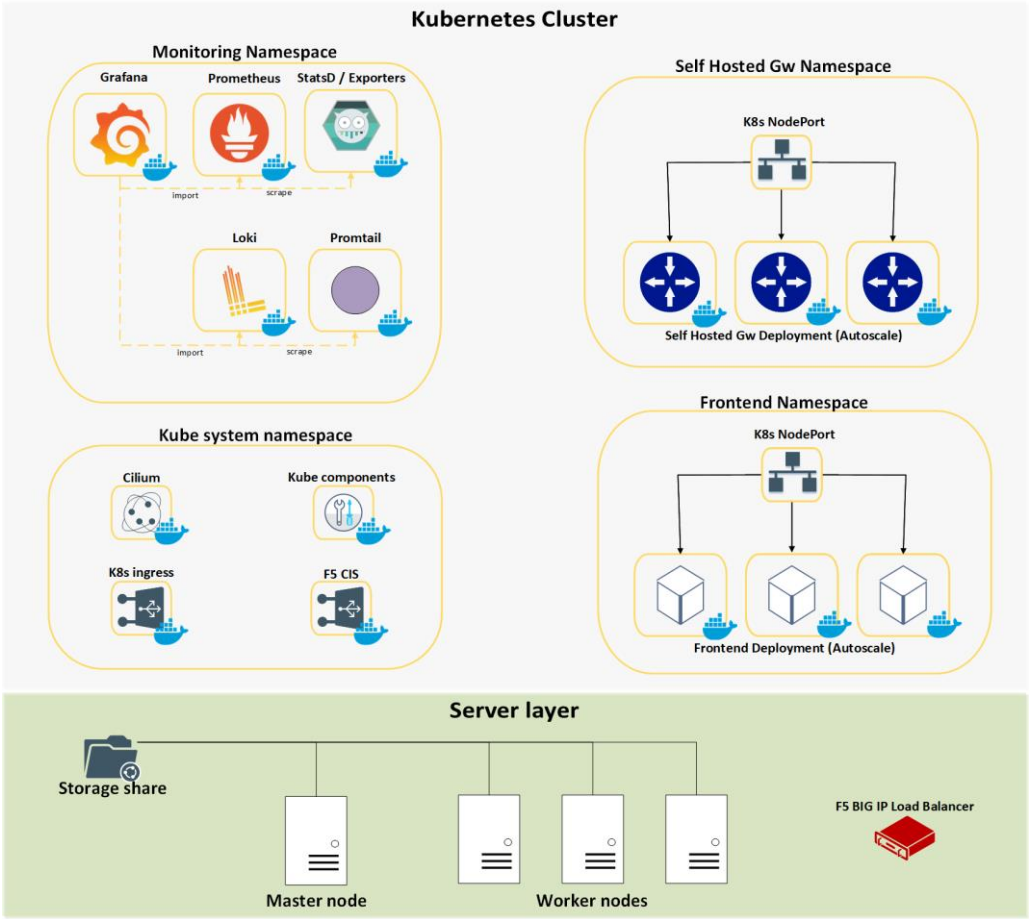
Keeps working w/o connection

- In case of loss of connectivity with Azure cloud, the self-hosted gateway keeps working without receiving configuration updates from the Azure APIM.



ARCHITECTURE

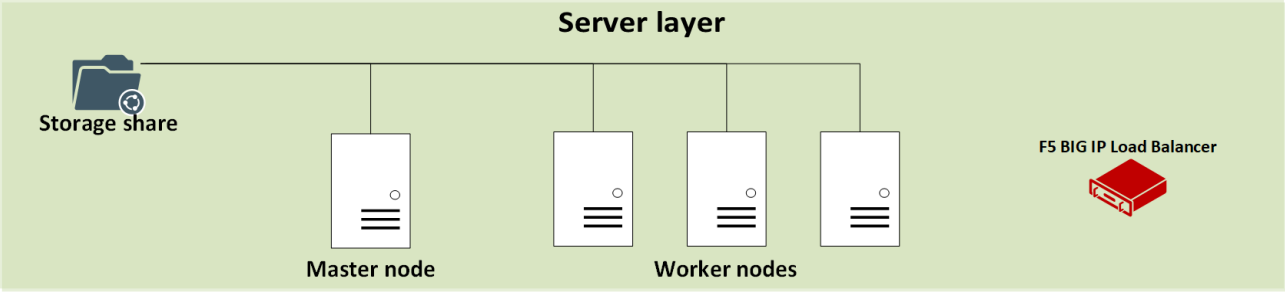
- Main components:
  - Server layer
  - Kubernetes layer
    - Networking
    - Observability stack
    - Application Stack



ON-PREMISES SIDE

SERVER LAYER

- 4 **RHEL** virtual machines from the same subnet.
- 1 external **file share** mounted on all the machines.
- 1 **F5 load balancer** distributing traffic across nodes.



ON-PREMISES SIDE

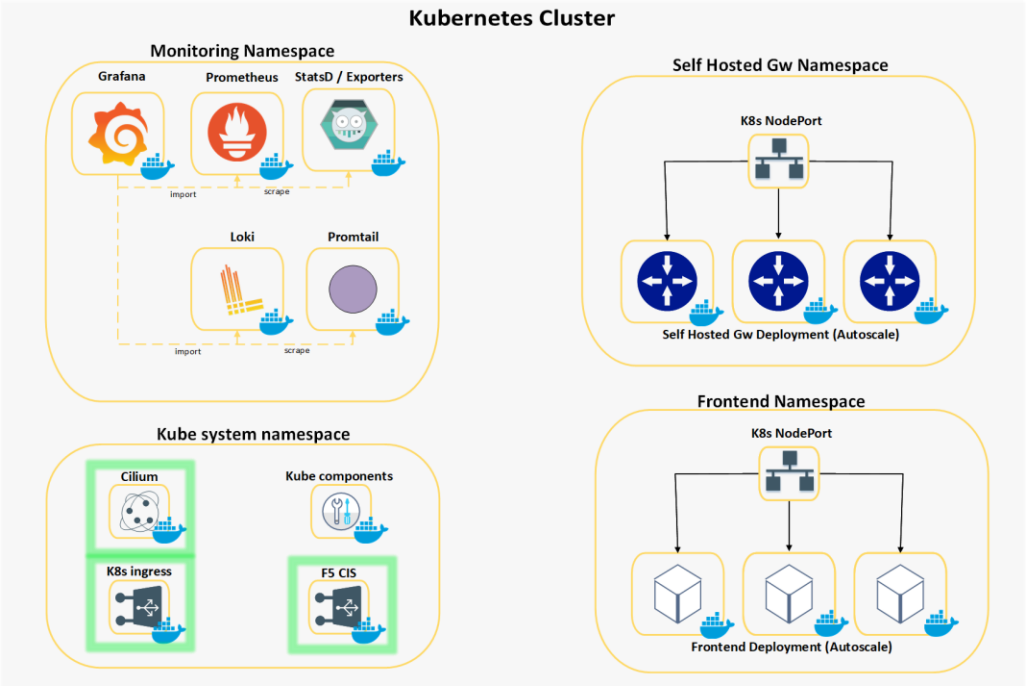
KUBERNETES LAYER - NETWORKING

Cilium CNI

- **Overlay** network model based on **eBPF**.
- Easy to install & use.

Ingress

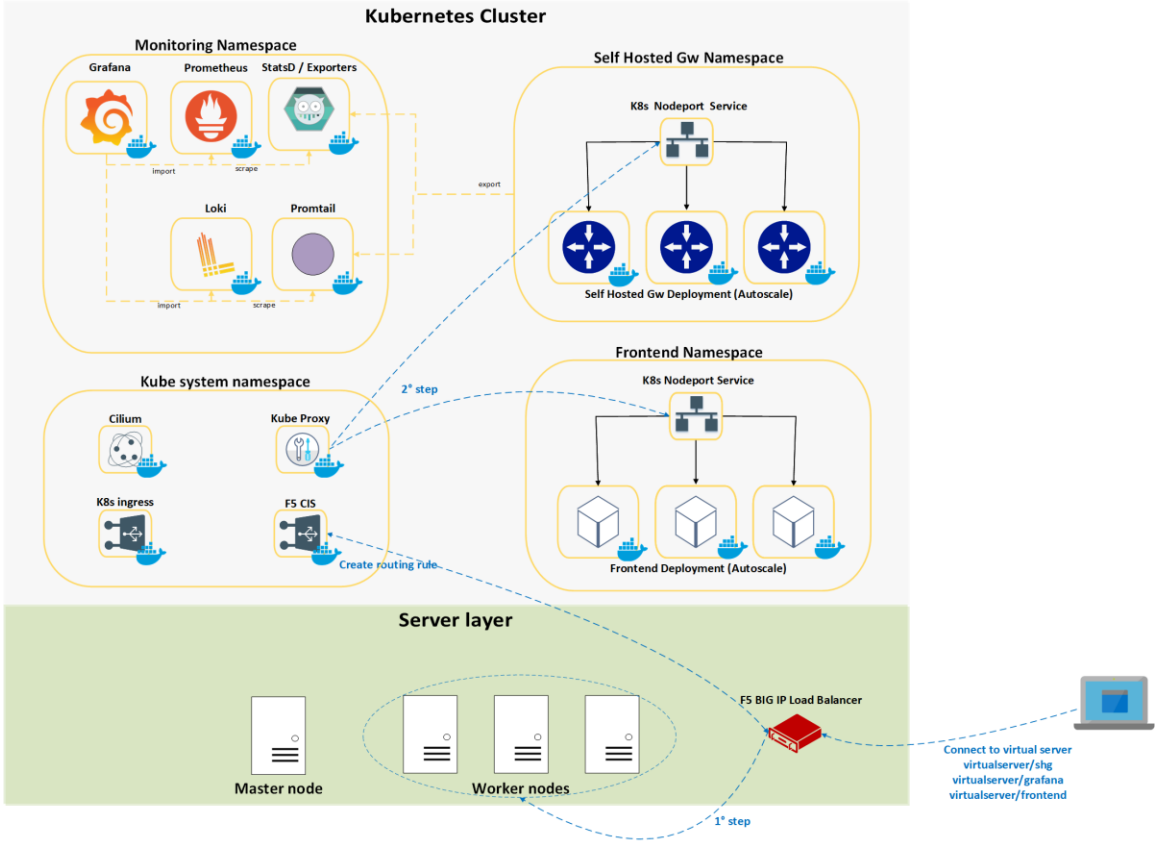
- **F5 Container Ingress Service (CIS)** in **NodePort** mode.
- CIS acts as an *ingress controller.*
- **Auto-configure** a virtual server on F5 when **ingress** rules are added.



ON-PREMISES SIDE

KUBERNETES LAYER - NETWORKING

- **NodePorts**: expose applications on each node on a given port.
- **Ingress manifest**: contains path based routing rules.
- **F5 CIS**: creates the corresponding rule on F5 each time the ingress is updated.
- **Kube-proxy:** performs the balancing across application replicas.



ON-PREMISES SIDE

KUBERNETES LAYER - OBSERVABILITY STACK

Grafana

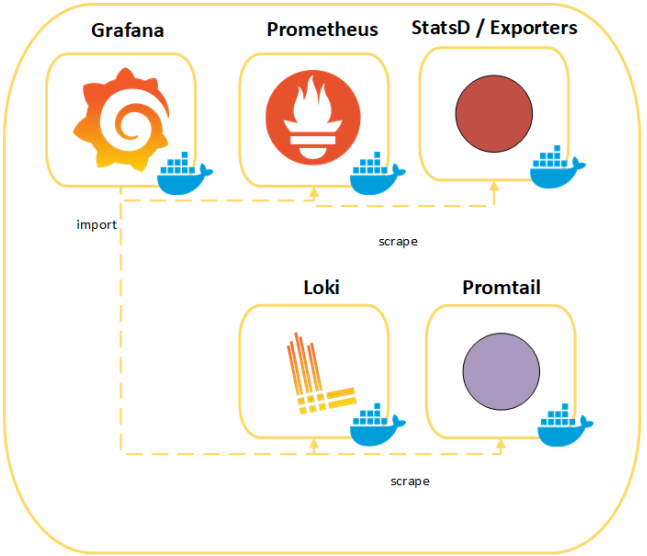
- Open-source **visualization** and **analytics** platform that unifies data sets into an interactive diagnostic workspace.

Loki

- Open-source **log aggregation** tool. Logs are **persisted** using the *storage share*.

Prometheus

- Open-source metrics-based **monitoring** and **alerting** tool.



ON-PREMISES SIDE

KUBERNETES LAYER - OBSERVABILITY STACK

Self-hosted gateway monitoring dashboard

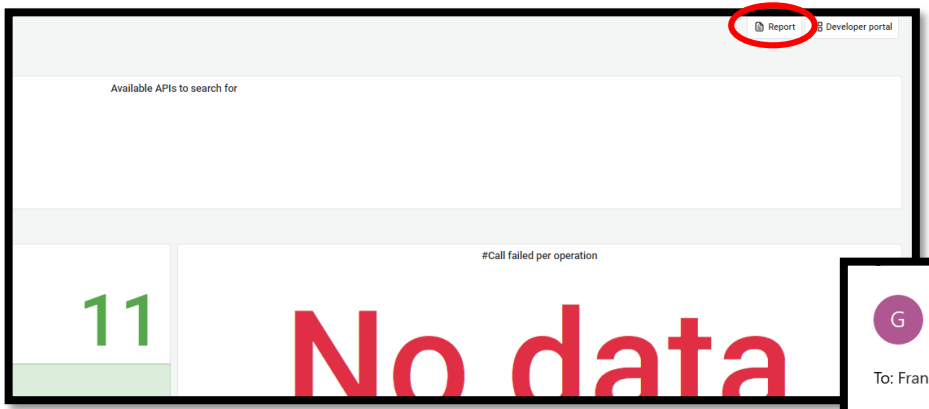
ON-PREMISES SIDE

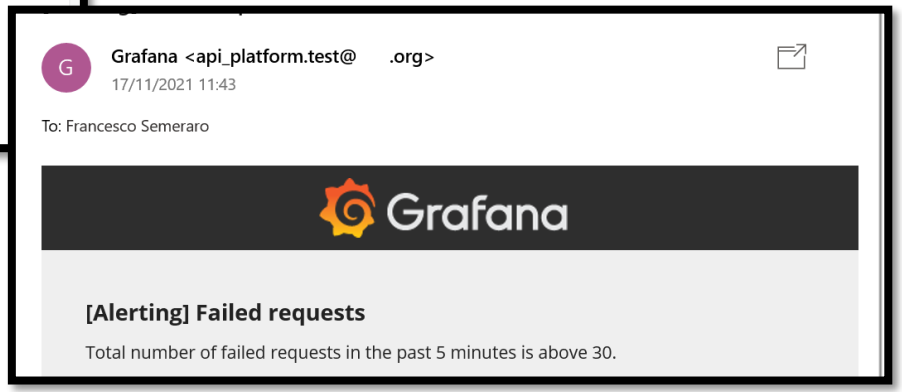
KUBERNETES LAYER - OBSERVABILITY STACK

Cluster monitoring dashboard

ON-PREMISES SIDE

KUBERNETES LAYER - OBSERVABILITY STACK





Alerting and reporting

Q&A

DEMO TIME

CONCLUSIONS

CONCLUSIONS

Qualitative

- Hybrid API Management combines the advantages of **cloud** and **on-premises** solutions.

- Azure API management with self-hosted gateway allows organizations to leverage Hybrid API management potential building a **feasible**, **reliable** and **agile** solution to address their business requirements.

Quantitative

- **Locust** distributed **testing framework** has been used to test the **load capabilities** of the self-hosted gateway (the cloud gateway comes at a predefined load capability depending on the tier):

vCPU	Memory (GiB)	# Workers	Max. requests (req/sec)	Gateway replicas
4	16	1	650	10
4	16	2	1100	10
4	16	3	1900	10
4	16	4	2500	10

THANK YOU

f.semeraro@reply.it

Politecnico di Torino

REPLY