

API MANAGEMENT IN THE CONTEXT OF HYBRID CLOUD

Francesco Semeraro | Cloud Engineer @Cluster Azure IT



Politecnico
di Torino



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.



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

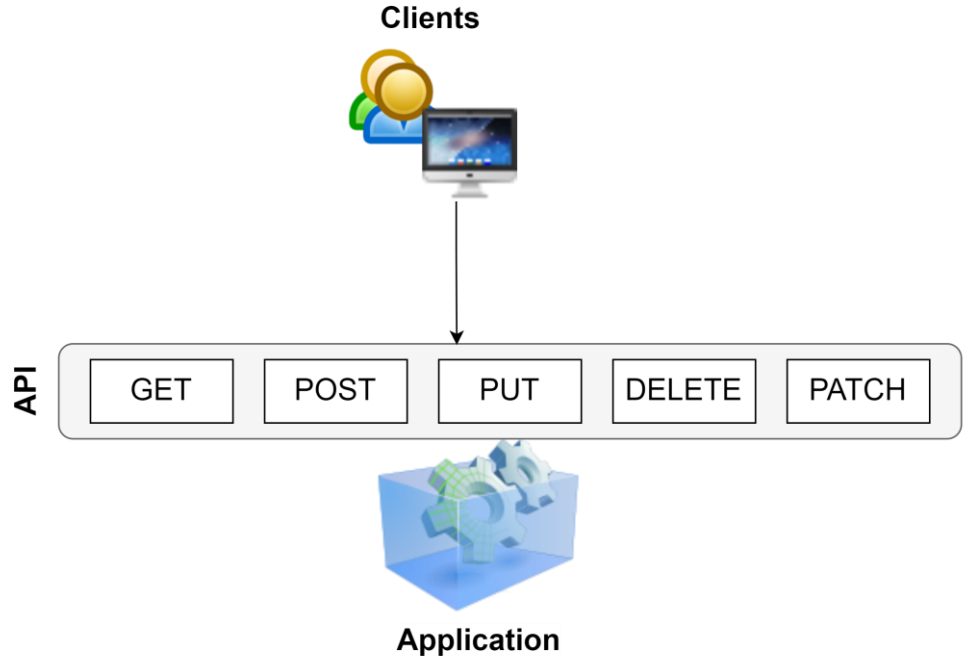
Demo

Conclusions

INTRODUCTION

WHAT IS AN API

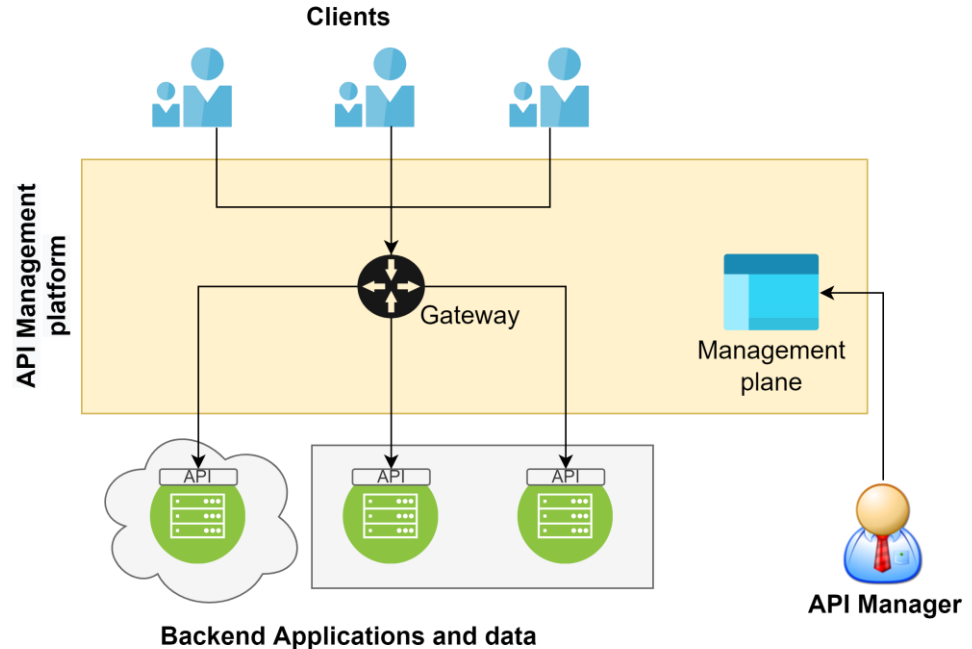
- **Application Programming Interfaces (APIs):** programming interfaces that allow users to access services, which can be simple **data** or **functionality**.
- Layer of **abstraction** between the **client** and the **services**, allowing the application developer (consumer) to disregard the coding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



API MANAGEMENT

DEFINITION

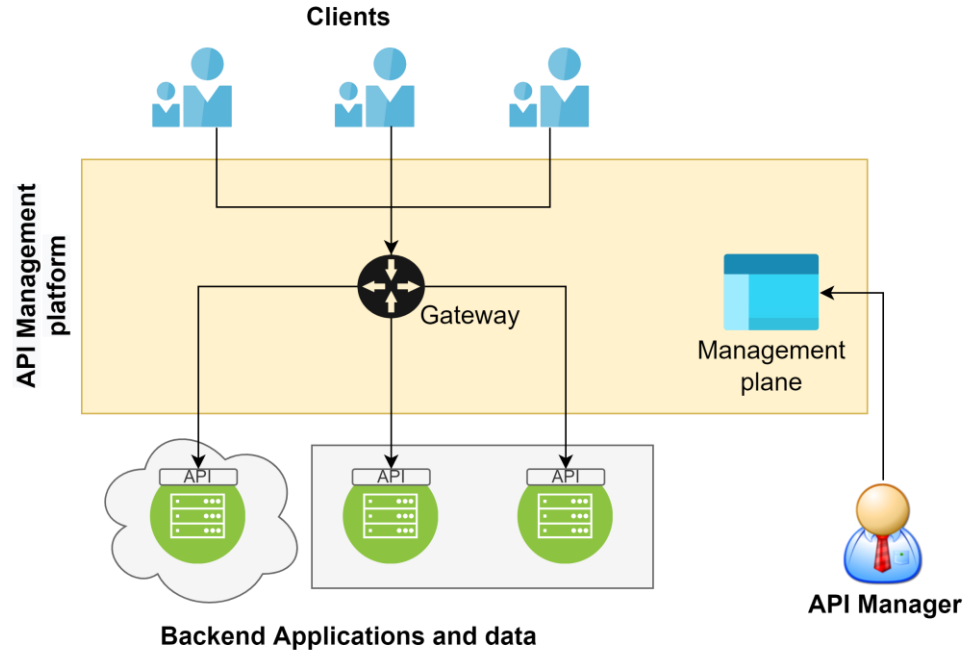
- **API management:** Exposes applications through well-defined **APIs**.
- **Backend apps and data** spread across multiple on-premises and cloud environments.
- Consistent and unified management of all API interfaces.



API MANAGEMENT

ADVANTAGES

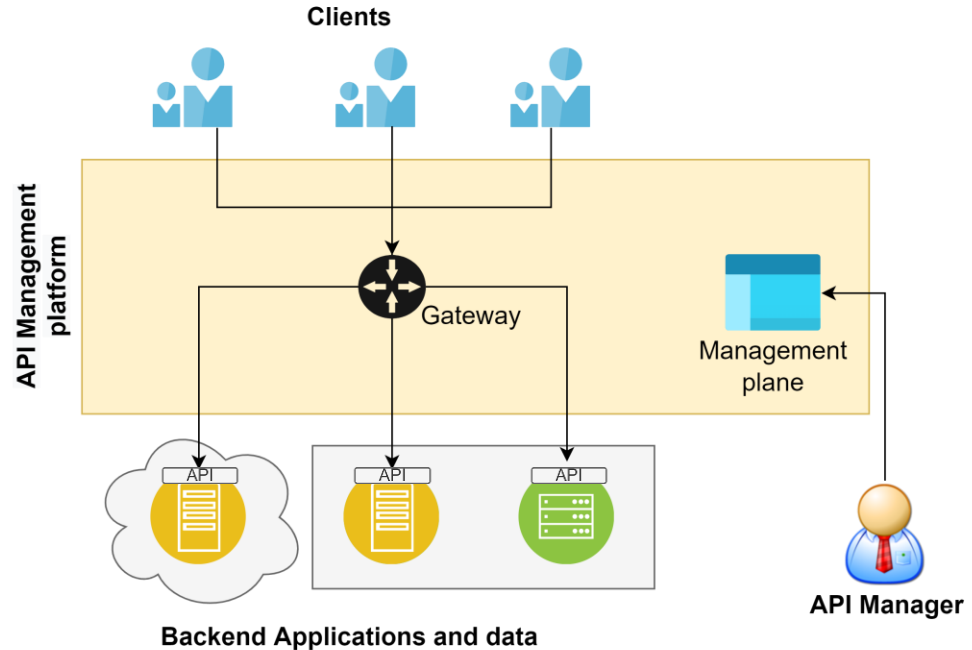
- **Self-service user onboarding:** clients can easily learn about APIs.



API MANAGEMENT

ADVANTAGES

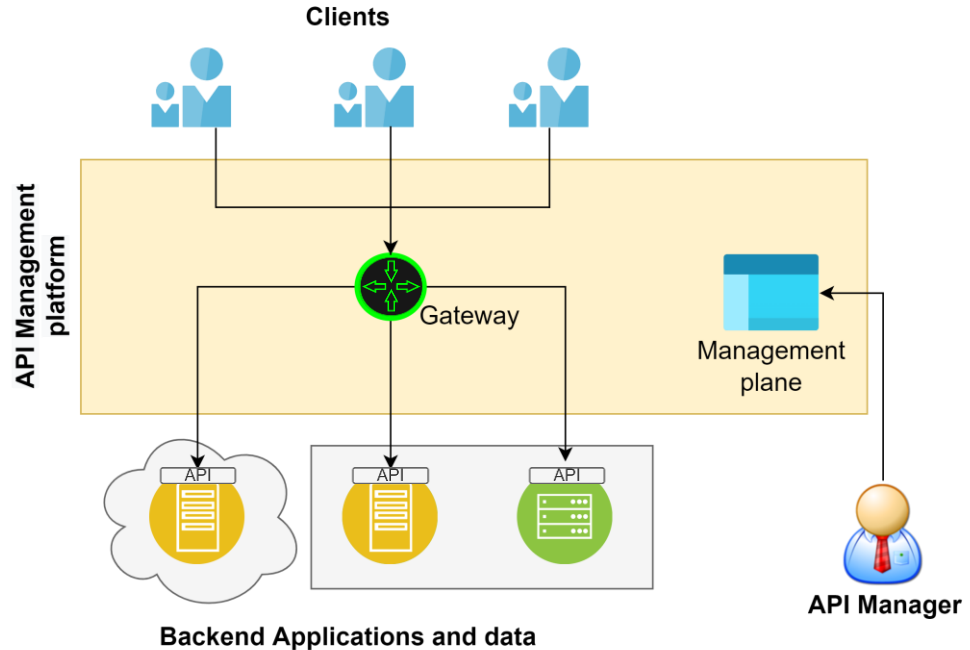
- **Self-service user onboarding:** clients can easily learn about APIs.
- **Façade hiding backends from clients:** you can move or re-architect backends without impacting clients.



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.

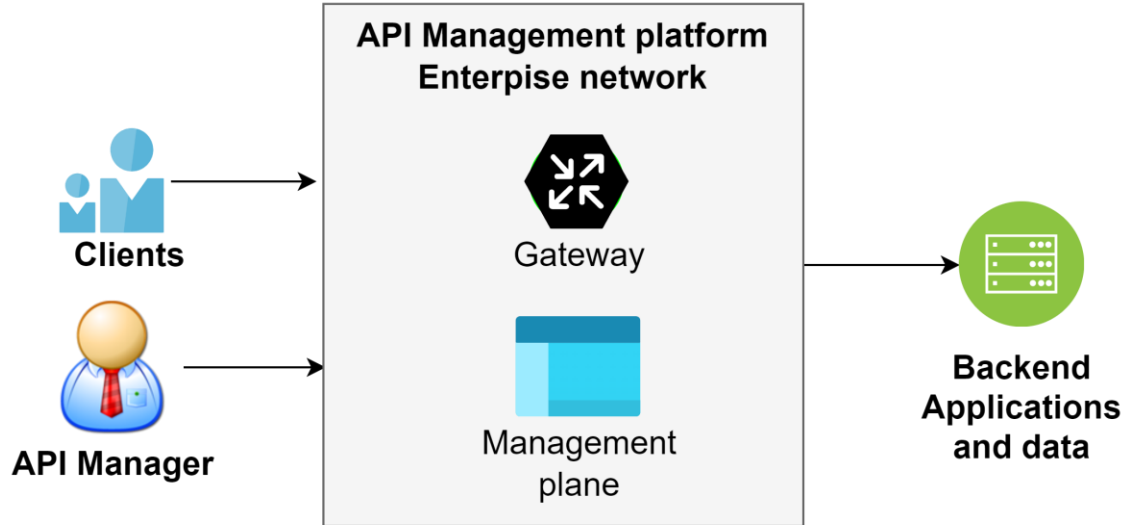


API MANAGEMENT

CLASSIC APPROACHES

On-prem hosted

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

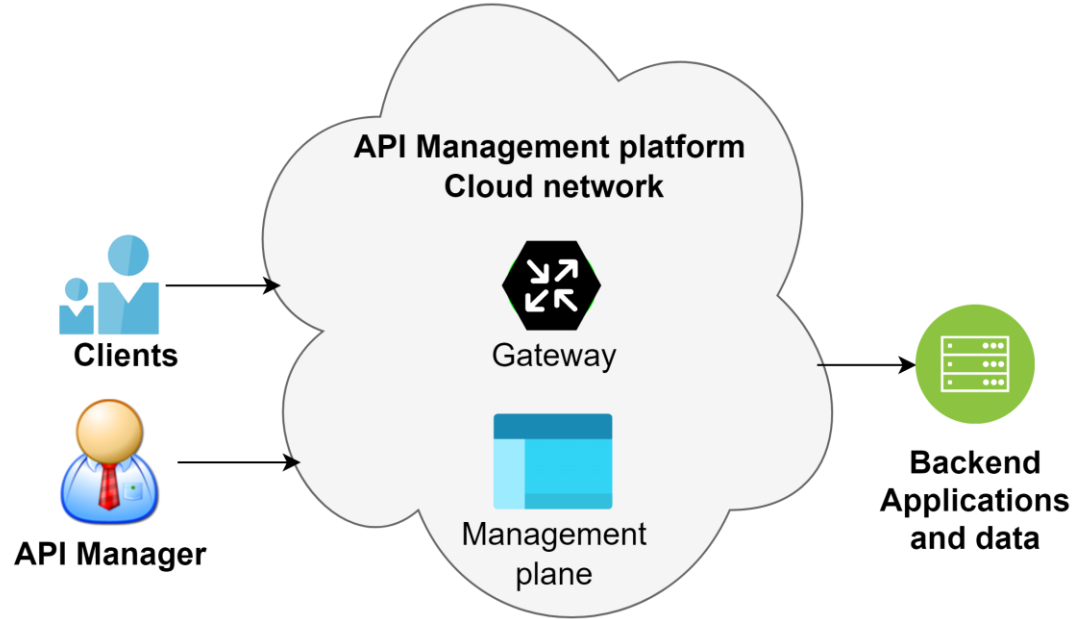


API MANAGEMENT

CLASSIC APPROACHES

Cloud-hosted

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



API MANAGEMENT

HYBRID API MANAGEMENT

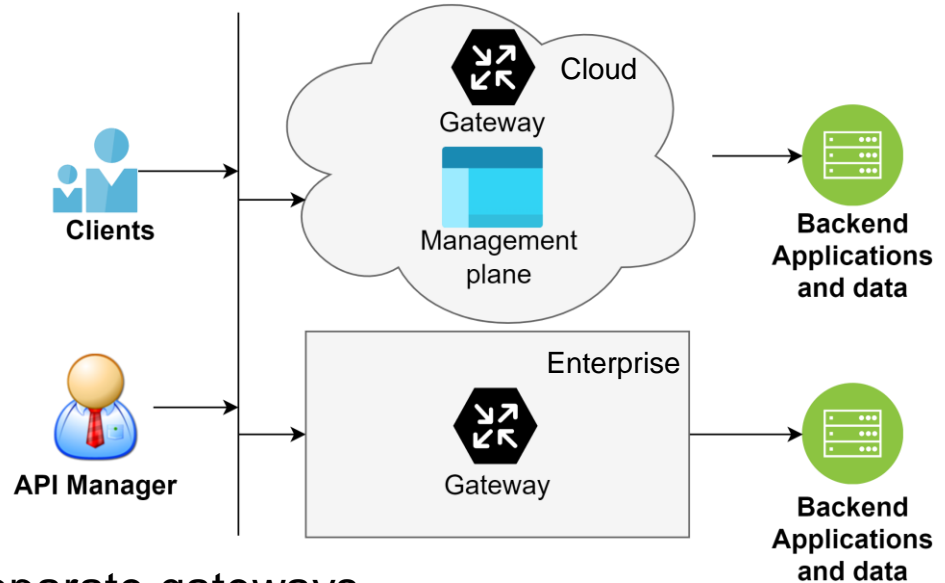
Hybrid API Management

Advantages:

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

Challenges:

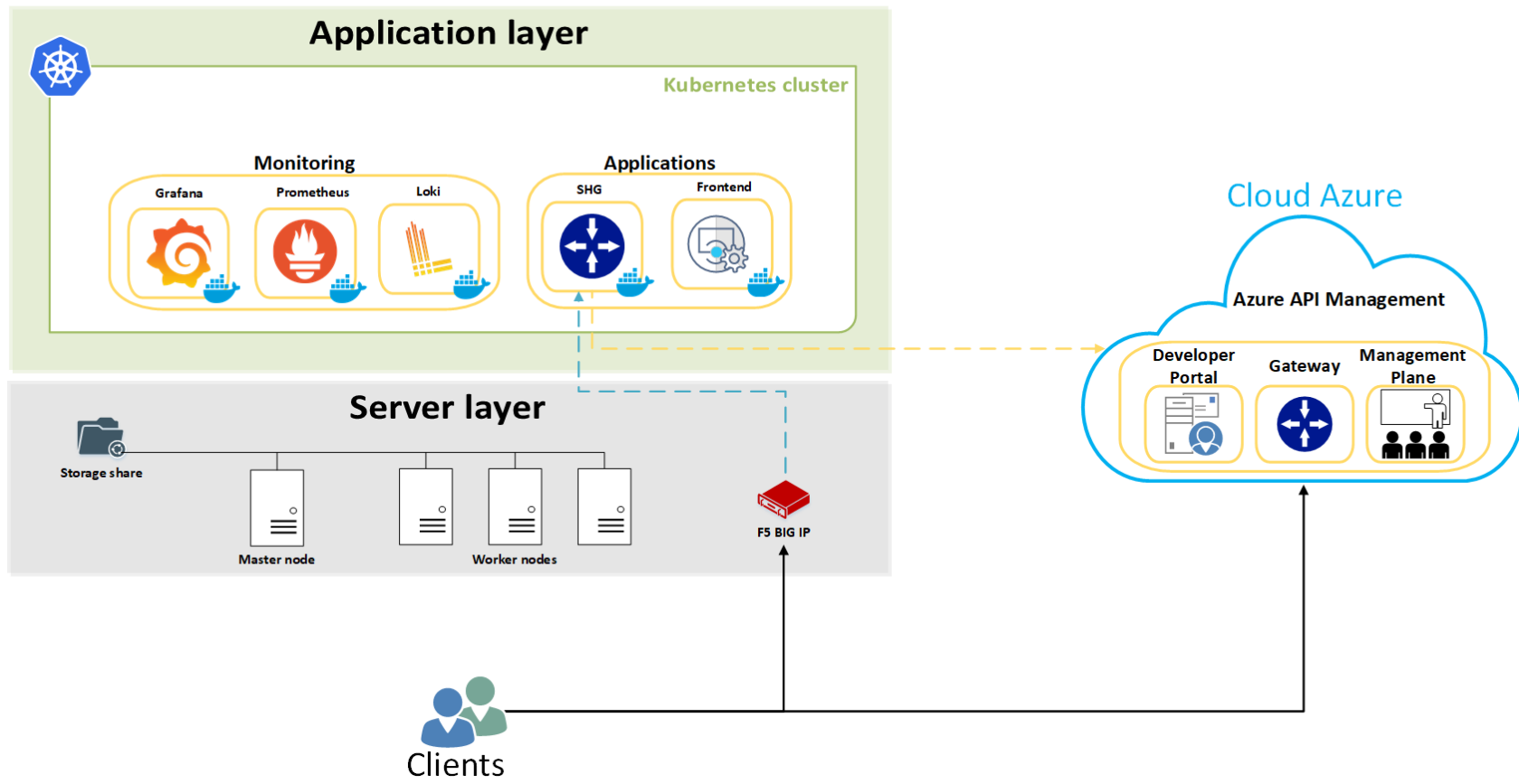
- Configuring and running many separate gateways.
- Avoid on-premises – cloud communication.
- It's **crucial to select the right API Management platform.**



HYBRID API MANAGEMENT SOLUTION

ARCHITECTURE

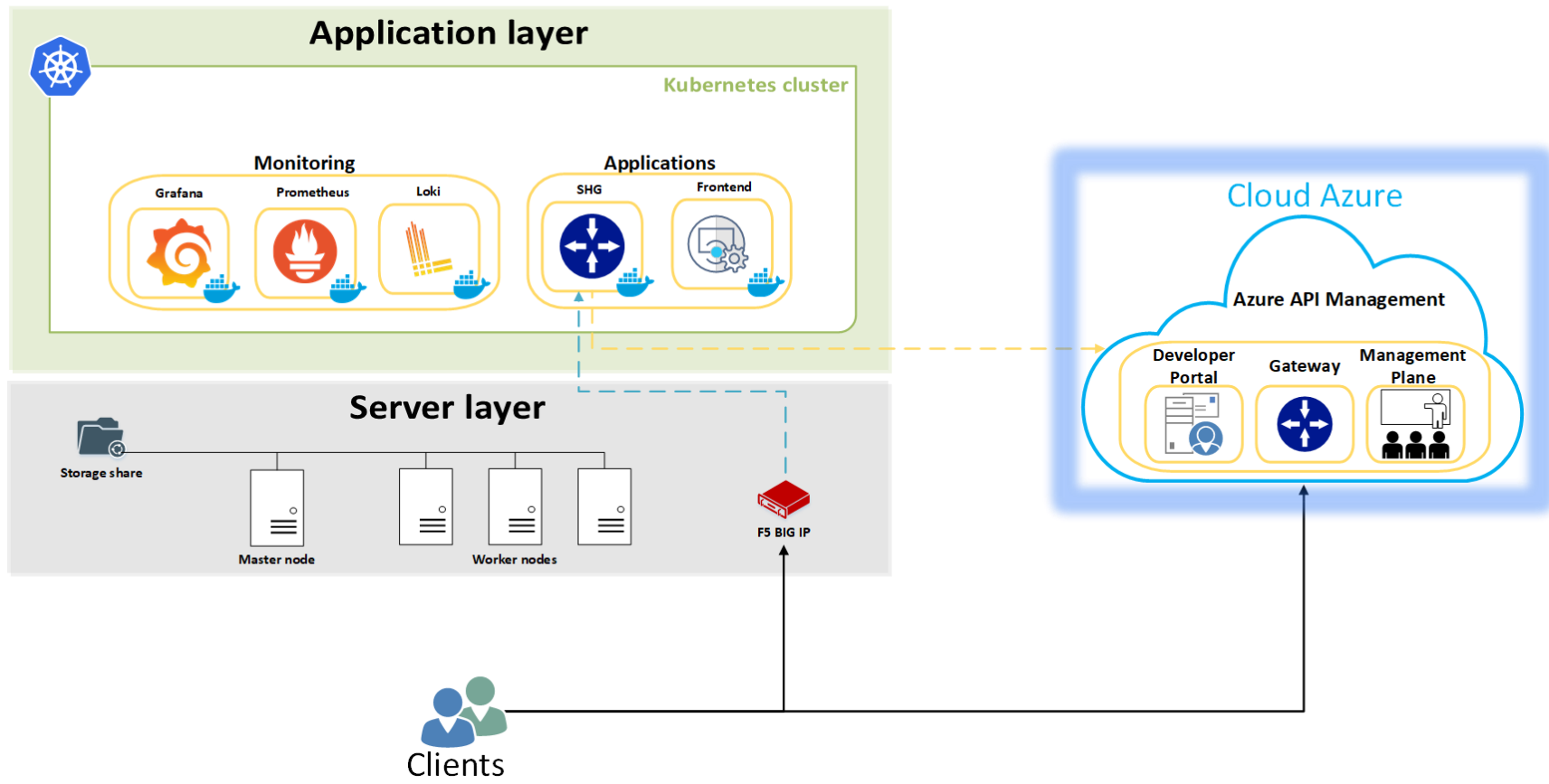
OVERVIEW



CLOUD SIDE



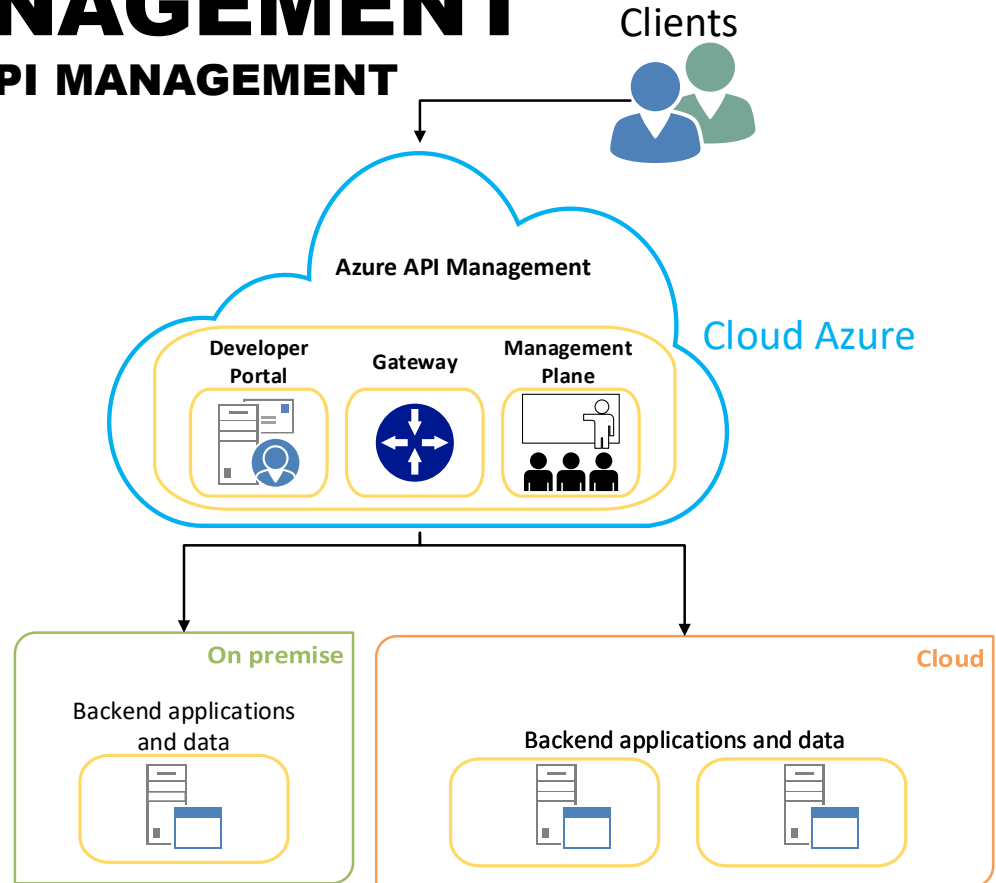
ARCHITECTURE



API MANAGEMENT

AZURE API MANAGEMENT

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



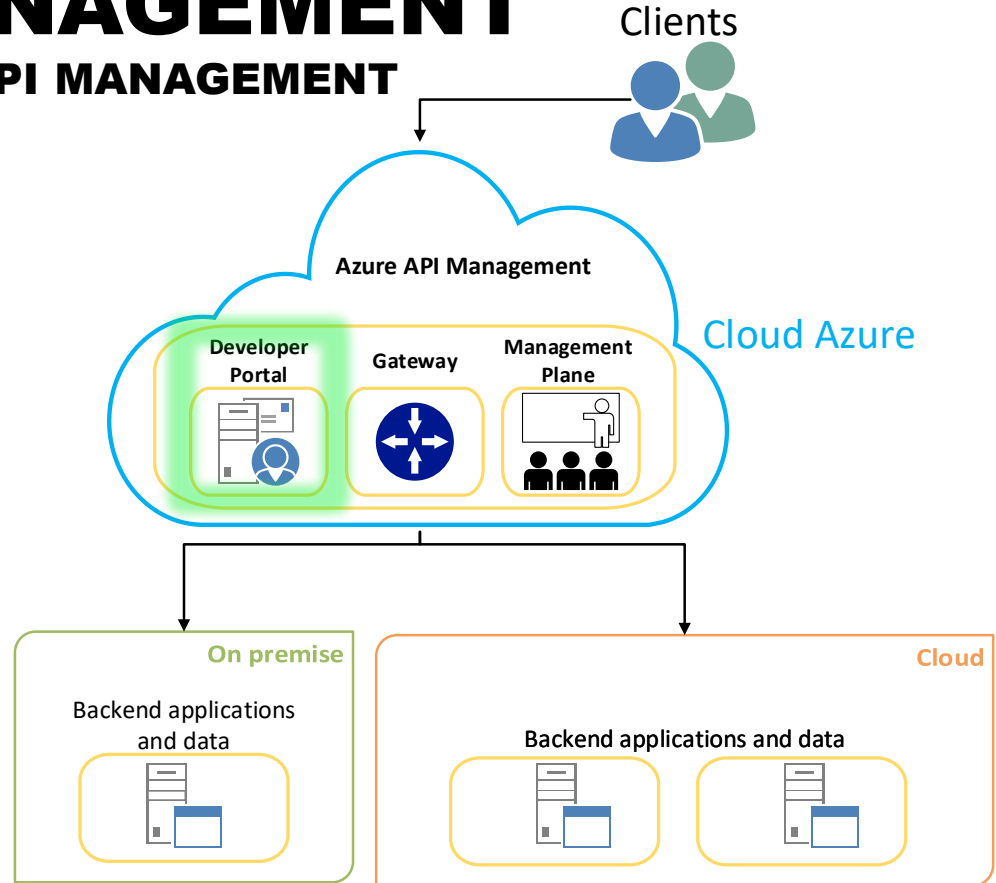
API MANAGEMENT

AZURE API MANAGEMENT

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

Developer portal

- Contains APIs documentation.



API MANAGEMENT

DEVELOPER PORTAL

Welcome to the developer portal!

All APIs documentation is available here

Sign up

Explore APIs

CloudAPI

API definition [Changelog](#)

This API is hosted on cloud and routed by the managed gateway.

InsertPost

Insert a new post.

CLOUD

Request

POST https://hybrid-apim01-dev.azure-api.net/cloudapi/posts

Request body

application/json

post

{ }

Name	Required	Type	Description
title	false	string	
body	false	string	
userid	false	integer	

Example

```
{  "title": "pollo",  "body": "seminar",  "userid": 5}
```

Headers

Content-Type	application/json	Remove
Cache-Control	no-cache	Remove

+ Add header

Body

Raw Binary

```
{  "title": "pollo",  "body": "seminar",  "userid": 5}
```

HTTP

HTTP request

Copy

POST https://hybrid-apim01-dev.azure-api.net/cloudapi/posts
HTTP/1.1

Content-Type: application/json
Cache-Control: no-cache

```
{  "title": "pollo",  "body": "seminar",  "userid": 5}
```

Send

API MANAGEMENT

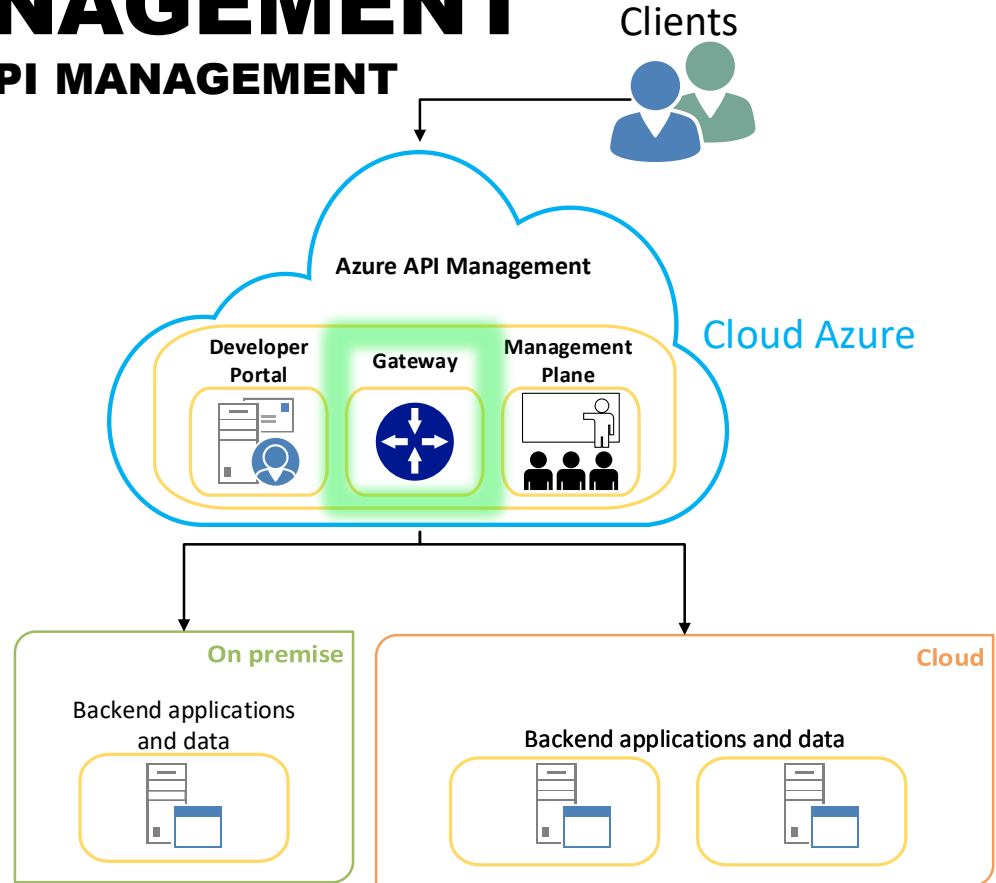
AZURE API MANAGEMENT

- 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

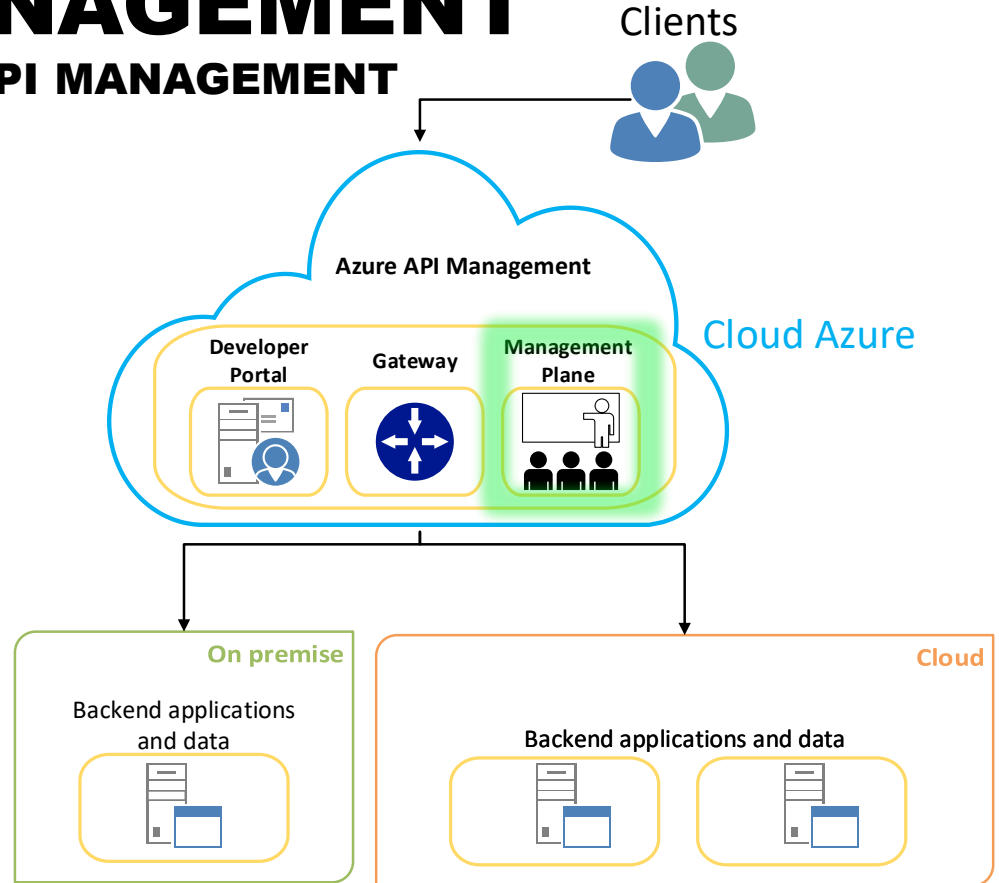
- Azure API management sits between clients and backend applications and data.

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

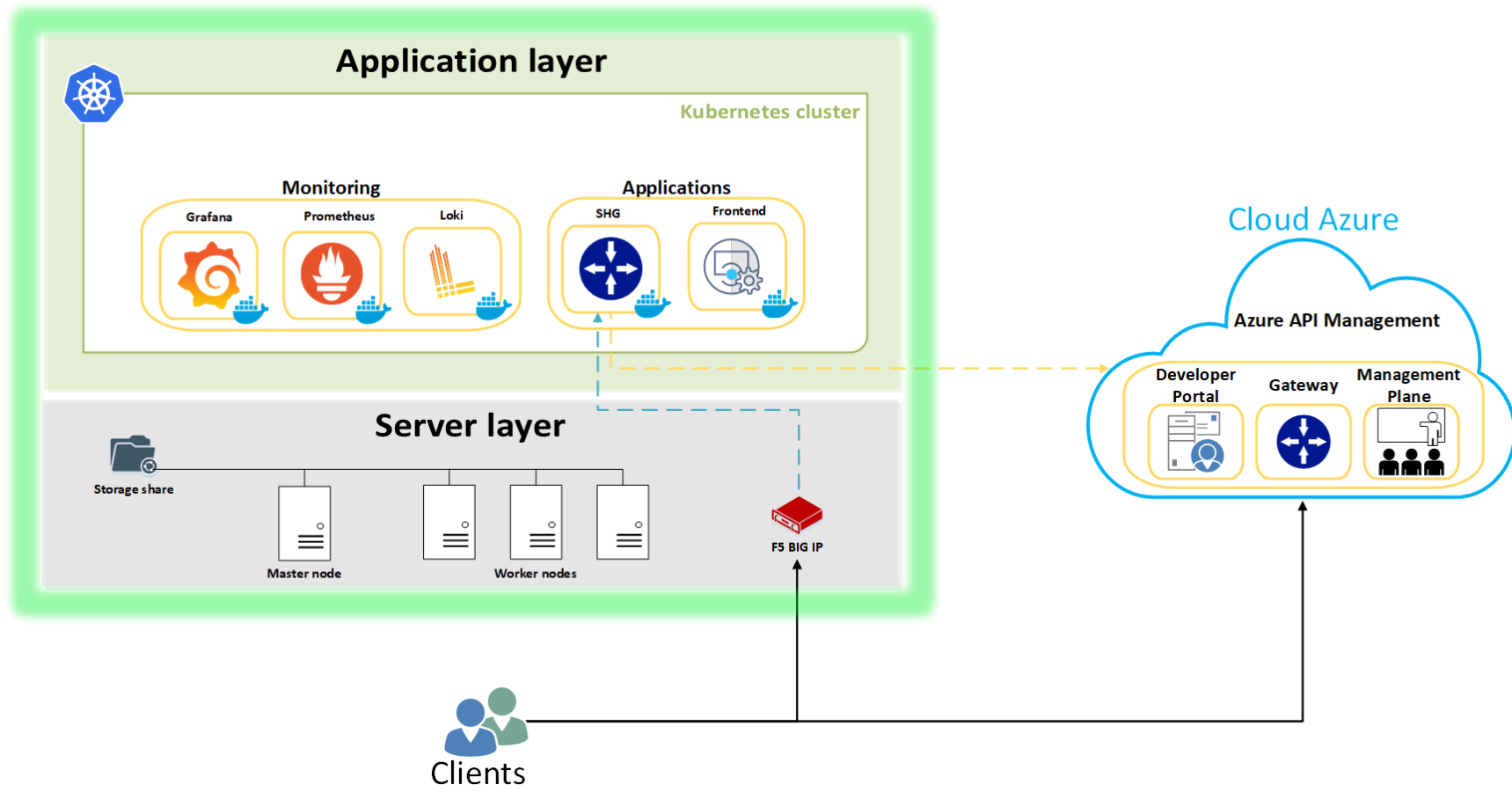
CONTROL POLICIES

```
1 <policies>
2   <inbound>
3     <rate-limit-by-key calls="5" renewal-period="1" counter-key="@context.Request.IpAddress" />
4     <rate-limit-by-key calls="100" renewal-period="60" counter-key="@context.Request.IpAddress" />
5     <rate-limit-by-key calls="5" renewal-period="1" counter-key="@context.Subscription?.Key ?? "anonymous" />
6     <rate-limit-by-key calls="100" renewal-period="60" counter-key="@context.Subscription?.Key ?? "anonymous" />
7     <ip-filter action="allow">
8       <address>13.66.201.169</address>
9     </ip-filter>
10  </inbound>
11  <backend>
12    <forward-request />
13  </backend>
14  <outbound />
15  <on-error />
16 </policies>
```

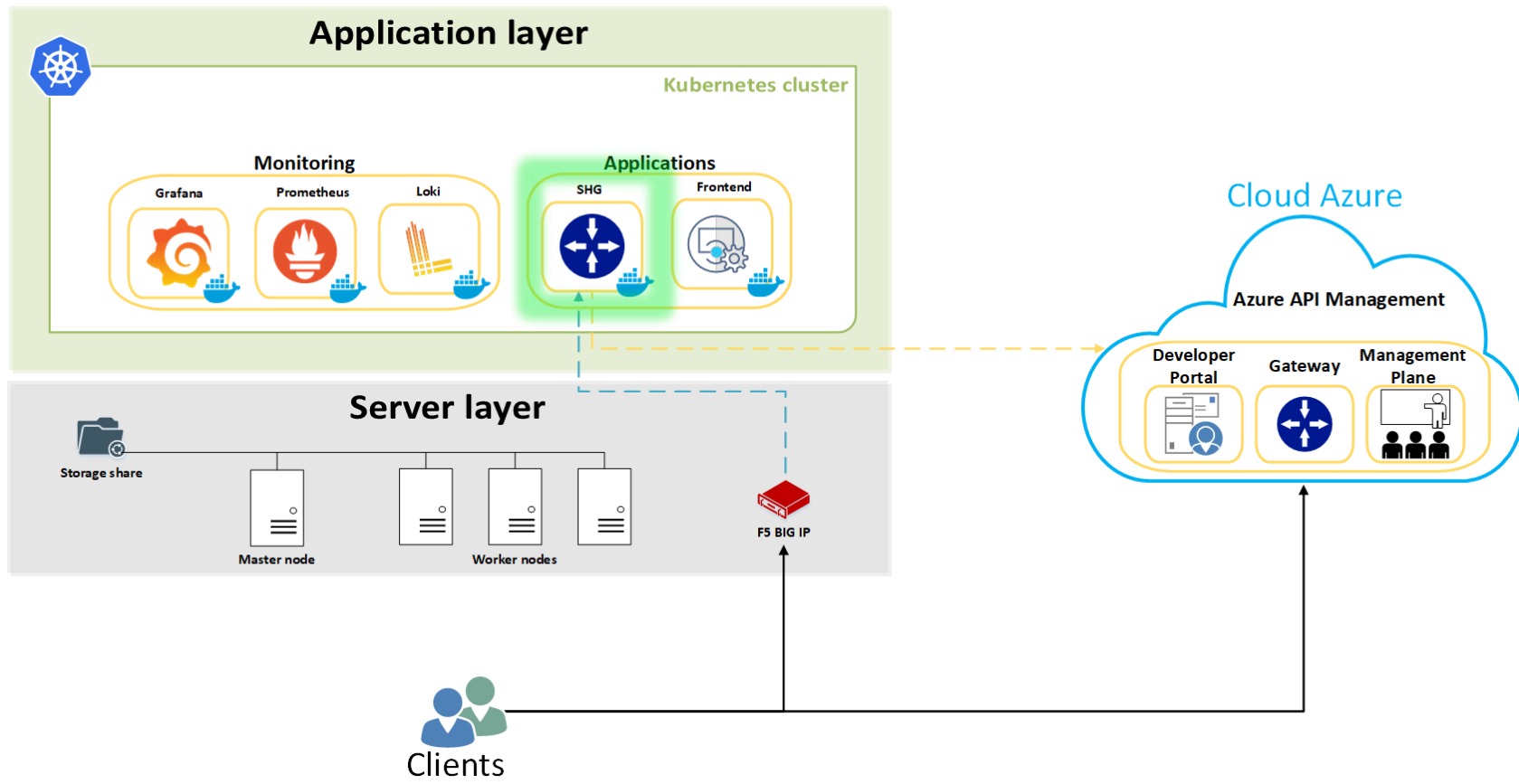
```
12 -->
13 <policies>
14   <inbound>
15     <base />
16     <set-backend-service base-url="CHANGEBACKENDURL" />
17     <authentication-certificate certificate-id="SSLCERTIFICATEID" />
18   </inbound>
19   <backend>
```

ON-PREMISES SIDE

ARCHITECTURE



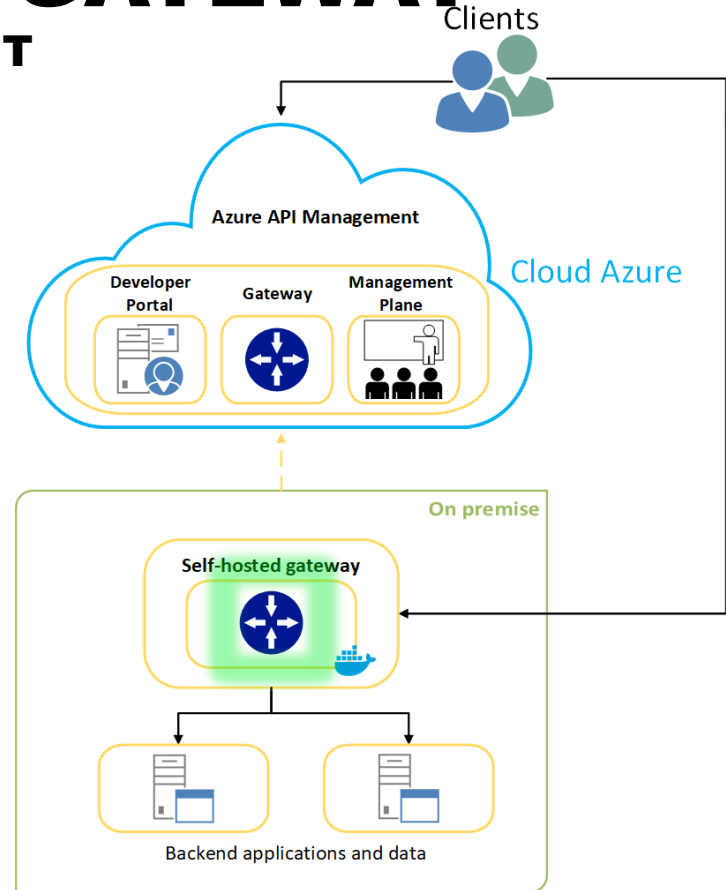
ARCHITECTURE



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.

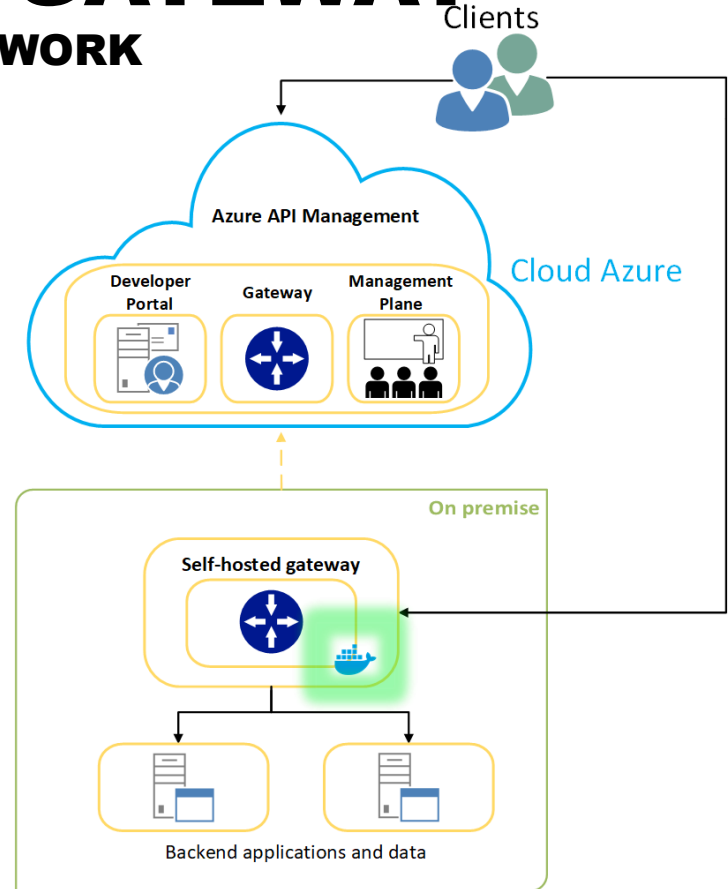


SELF HOSTED GATEWAY

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.



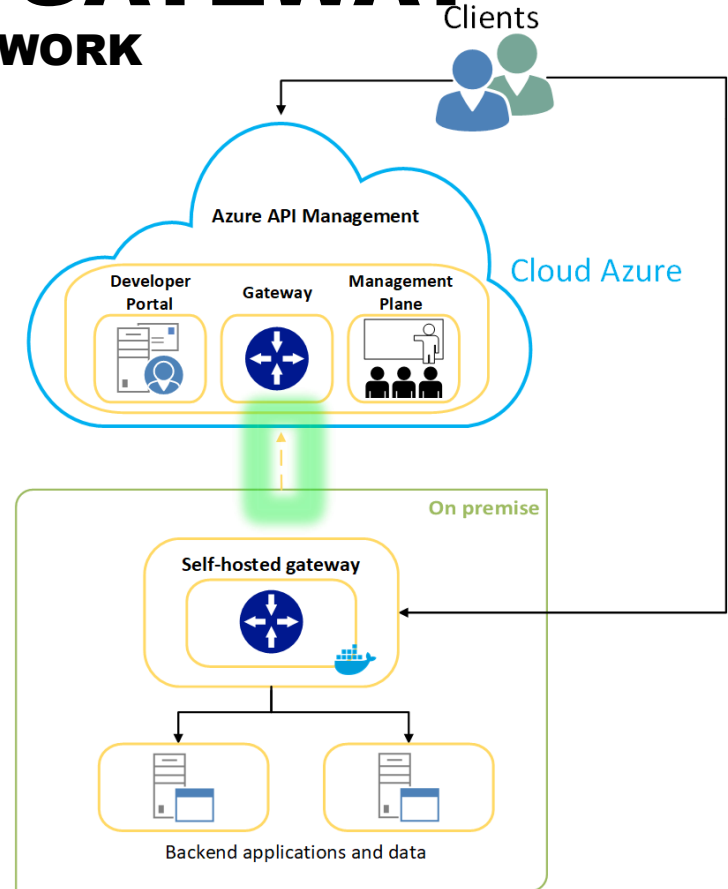
SELF HOSTED GATEWAY

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

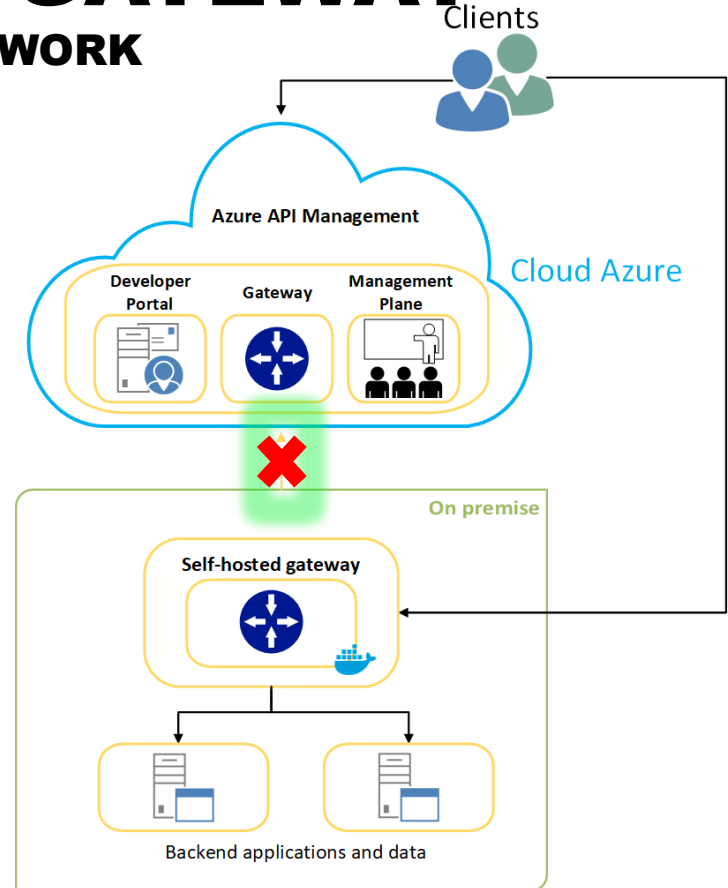
HOW DOES IT WORK

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:

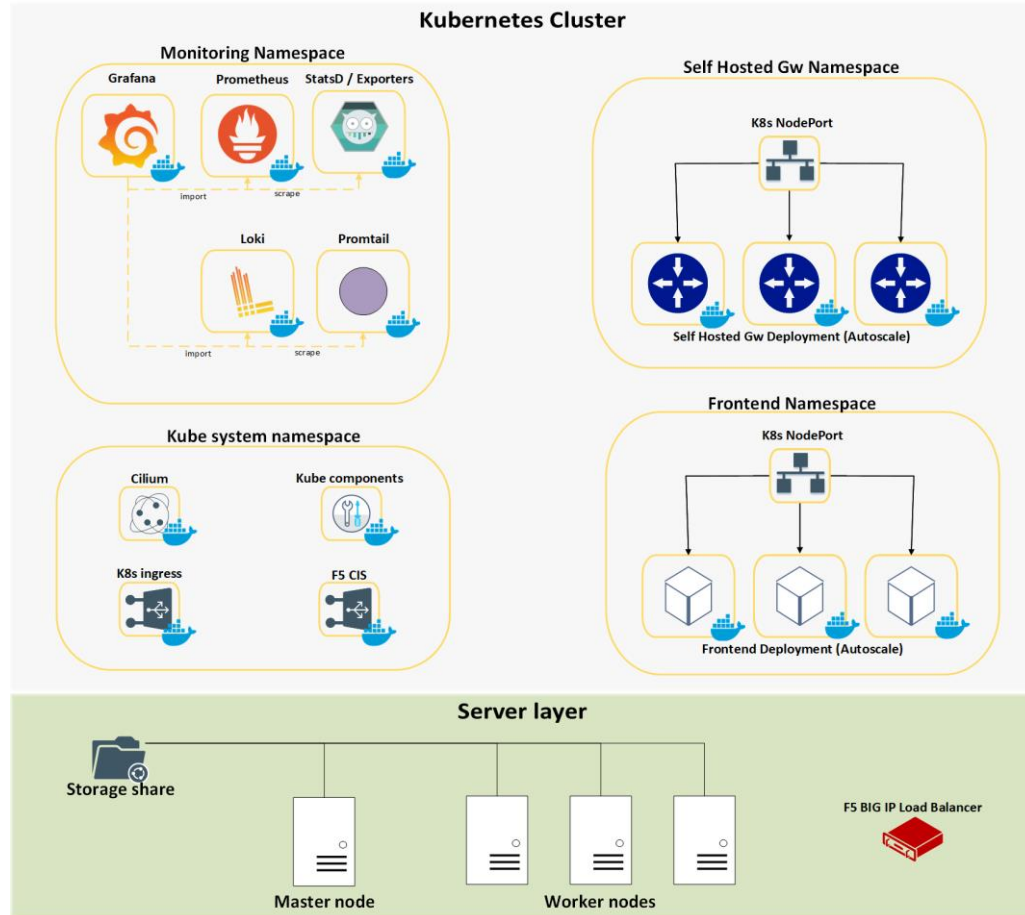
Server layer

Kubernetes layer

Networking

Observability stack

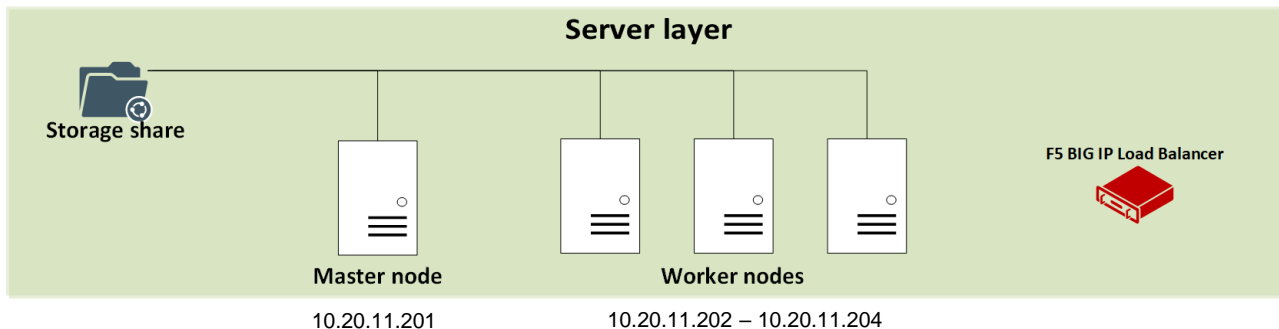
Application Stack



ON-PREMISES SIDE

SERVER LAYER

- 4 **RHEL** virtual machines from the 10.20.11.0/24 subnet.
- 1 external **file share** mounted on all the machines.
- 1 **F5 load balancer** distributing traffic across nodes.



ON-PREMISES SIDE

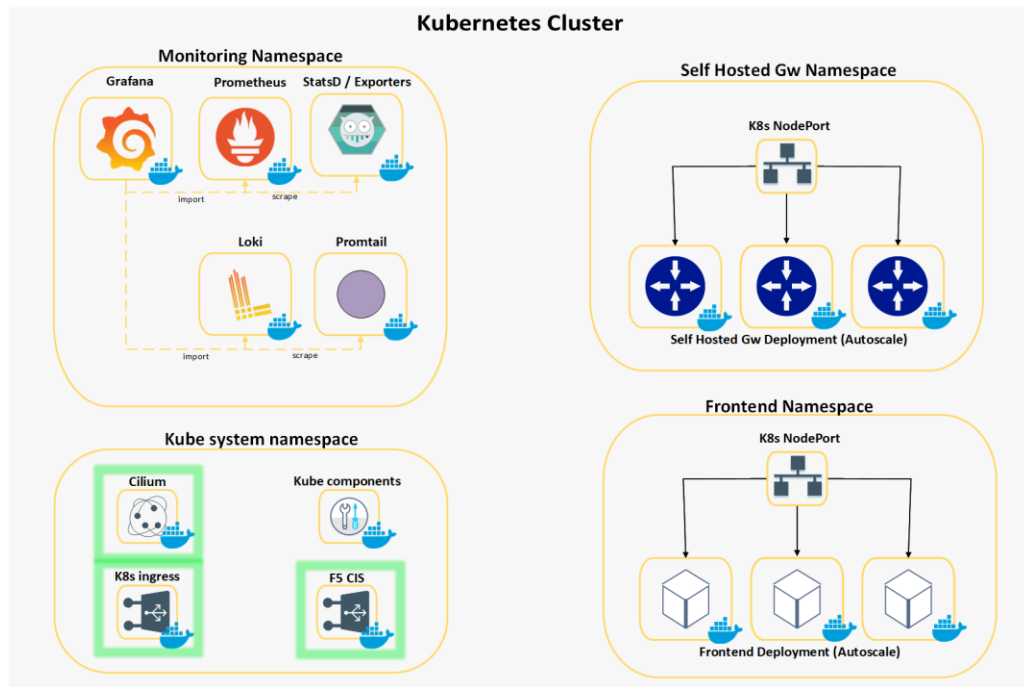
KUBERNETES LAYER - NETWORKING

Cilium CNI

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

Ingress

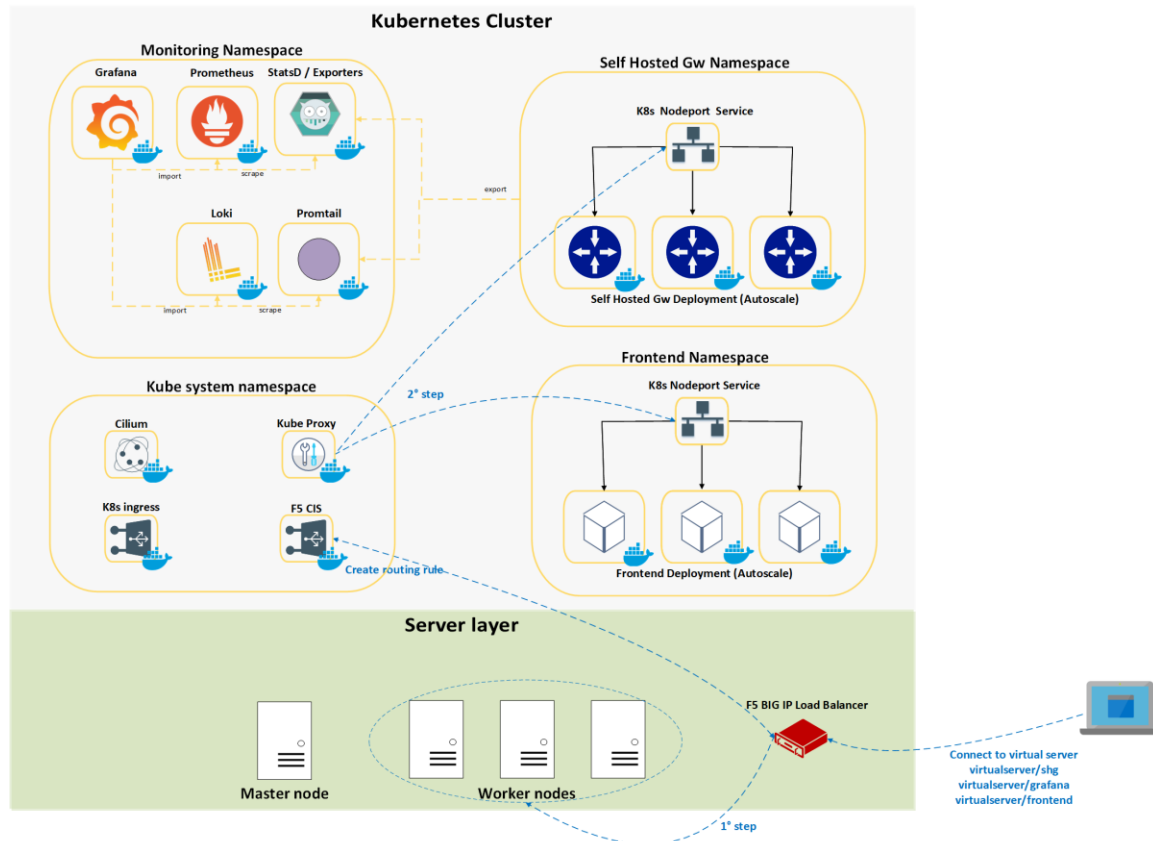
- **F5 Container Ingress Service (CIS)** in **NodePort** mode.
- CIS acts as an *ingress controller*.
- **Auto-configure** a virtual server on F5 when **ingress** rules are added.



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

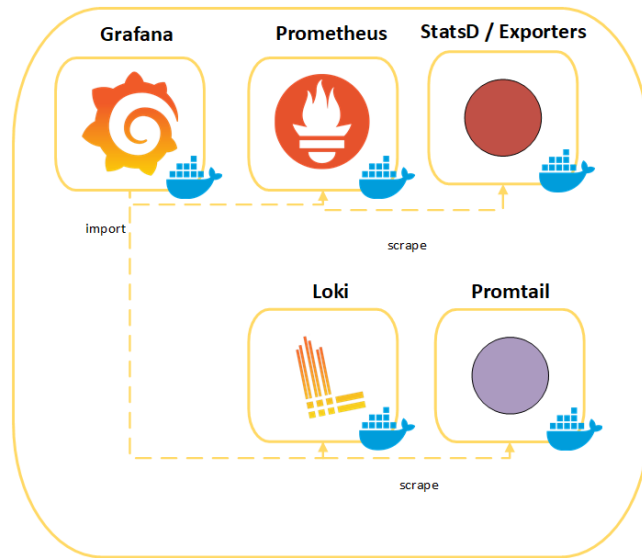
- Open-source **visualization** and **analytics** platform that unifies data sets into an interactive diagnostic workspace.

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

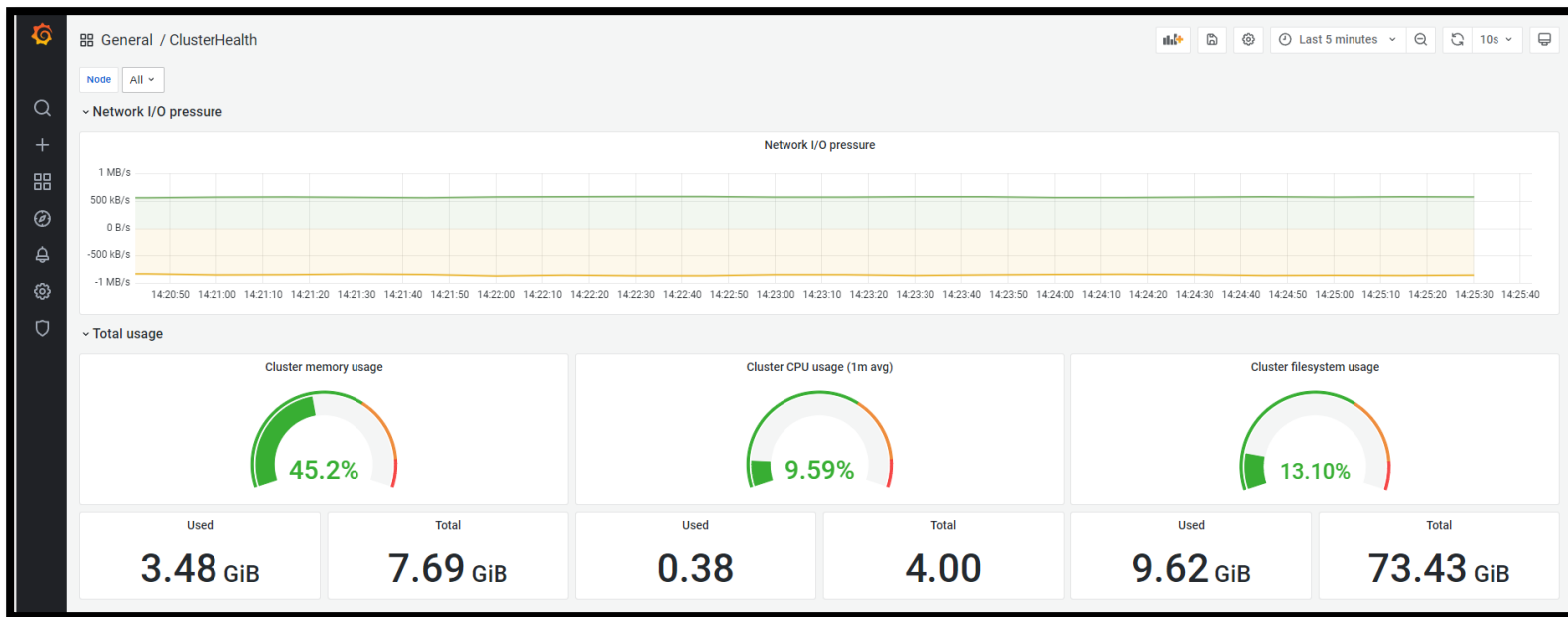
KUBERNETES LAYER - OBSERVABILITY STACK



*Self-hosted gateway
monitoring dashboard*

ON-PREMISES SIDE

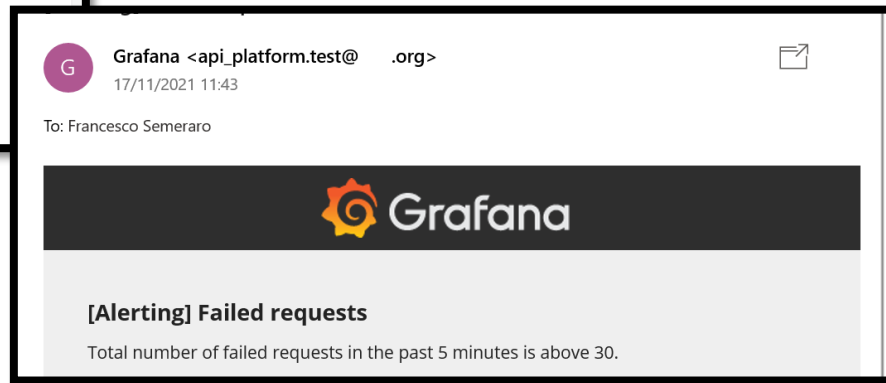
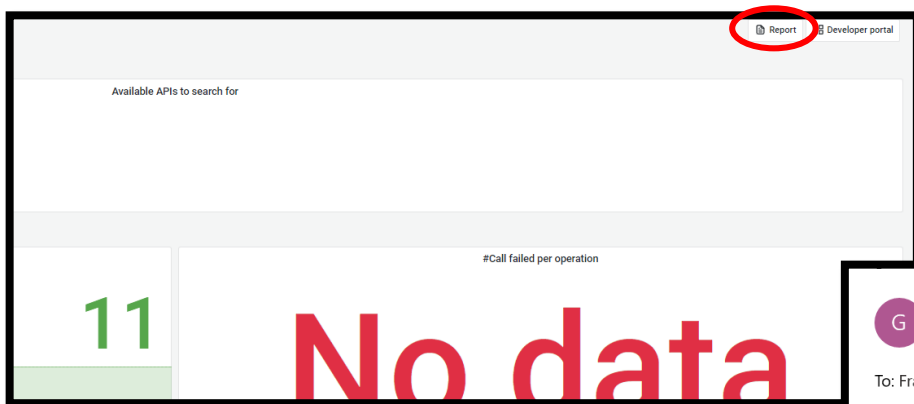
KUBERNETES LAYER - OBSERVABILITY STACK



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

