

Onboarding AKS to Azure Event Grid

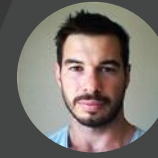
2020 Internship Project



Aman Jaiman
Software Engineer Intern



Ganeshkumar Ashokavardhanan
Software Engineer Intern



Stephane Erbrech
Manager

Motivation

Background and Approach

Azure Kubernetes Service

- Managed Kubernetes service by Microsoft
- Frequent minor and patch updates of Kubernetes

Problem

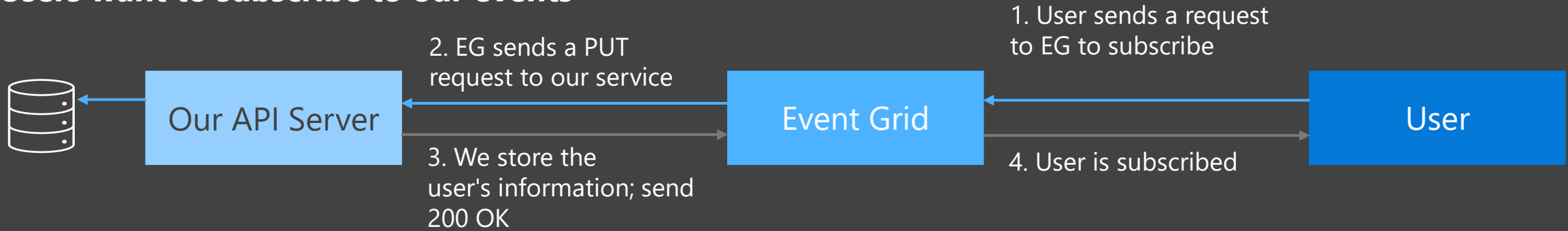
- Users cannot reliably act on events emitted by AKS Resource Provider
- Users want to track events such as availability of new Kubernetes versions for upgrades

Approach

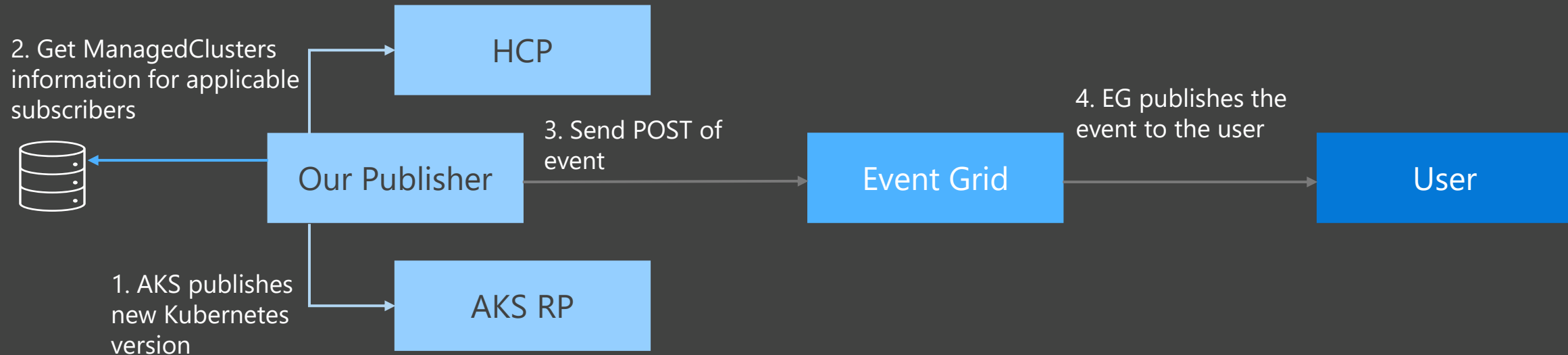
- Implement infrastructure needed to onboard AKS as an Event Grid publisher
- Create reusable artifacts for other services that onboard onto Event Grid

Architecture

Users want to subscribe to our events



Publishing an event



REST API Server

Onboarding to Event Grid

For Event Grid to send requests to AKS

- Followed the many steps for Event Grid Onboarding
- Handles four requests: PUT, GET, GET ALL, DELETE

Challenges

- Handle errors, headers and many specifications for Event Grid
- Handle packages in Go appropriately – many dependencies
- HTTP request passed through many functions (low level)

Protobufs and gRPC

Using an extensible framework

Motivation

- Onboarding process is standard yet time consuming
- Create reusable artifacts that any Azure service can use
- AKS is also migrating to gRPC for standardization and speed



gRPC Migration

- gRPC: open source, high performance RPC framework
- Use HTTP/gRPC transcoding to handle HTTP requests from Event Grid
- Use bufbuild/buf container additional proto support
- Server implementation handles gRPC requests and calls same functions for earlier business logic

Protobufs and gRPC

- Define the Event Grid API in a .proto file
- Generate types, constructors, server and client stubs in 7+ major languages (Go, Java, Python, C#)
- **Can be reused by any Azure service onboarding onto Event Grid**

```
service EventGridService {  
    rpc GetEventGridFilter (GetEventGridFilterRequest)  
    returns (GetEventGridFilterResponse) {}  
  
    rpc GetAllEventGridFilters (GetAllEventGridFiltersRequest)  
    returns (GetAllEventGridFiltersResponse) {}  
}  
  
message GetEventGridFilterRequest{  
    string subscription_id = 1;  
    string resource_group_name = 2;  
    string resource_name = 3;  
    string filter_name = 4;  
}  
  
message GetAllEventGridFiltersResponse {  
    repeated GetEventGridFilterResponse filters = 1;  
}
```

HTTP/gRPC transcoding using a YAML file

Generate gateway/reverse proxy using this file!

```
http:
```

```
  rules:
```

- selector: eventgrid.services.v1.EventGridService.GetEventGridFilter
- get: "/subscriptions/{subscription_id}/resourceGroups/{resource_group_name}/providers/..."
- selector: eventgrid.services.v1.EventGridService.GetAllEventGridFilters
 get: "/subscriptions/{subscription_id}/resourceGroups/{resource_group_name}/providers/..."
- selector: eventgrid.services.v1.EventGridService.PutEventGridFilter
 put: "/subscriptions/{subscription_id}/resourceGroups/{resource_group_name}/providers/..."
 body: "body"
- selector: eventgrid.services.v1.EventGridService.DeleteEventGridFilter
 delete: "/subscriptions/{subscription_id}/resourceGroups/{resource_group_name}/providers/..."

Postman: Local Testing

Motivation

- Calling API Server locally with to test routes
- Writing Postman tests to ensure we follow the Event Grid requirements

PUT:

```
localhost:{{port}}/subscriptions/{{subscriptionId}}/resourceGroups/{{rgName}}  
/providers/Microsoft.ContainerService/ManagedClusters/{{resourceName}}/eventG  
ridFilters/{{egFilterName}}
```

```
pm.test("Return 200", function () {  
    pm.response.to.have.status(200);  
});
```

```
pm.test("Request body in response", function () {  
    pm.expect(JSON.stringify(JSON.parse(responseBody).properties.filter)).to.  
include(JSON.stringify(JSON.parse(request.data).properties.filter))  
});
```


Postman: Documentation

Motivation

- Create Postman documentation that can be easily followed and tested by other Azure services
- Provide code samples for quick testing of a new API Server
 - Samples provided in 25+ languages and flavors



The screenshot shows a Postman API endpoint documentation page for a GET request. The endpoint is titled "GET GetEventGridFilter" and has a "Comments 0" button. The URL is "localhost:8080/subscriptions/subscriptionId/resourceGroups/resourceGroupName/providers/Microsoft.ContainerService/ManagedClusters/resourceName/eventGridFilters/egFilterName". Below the URL, there are two sections for tests: "Tests for GET" and "CORRECT RESPONSE CODE". The first test is "CORRECT RESPONSE CODE" with the code:

```
pm.test("Return 200", function () {  
    pm.response.to.have.status(200);  
});
```

 The second test is "CORRECT SUBSCRIPTION RETRIEVED" with the code:

```
pm.test("Has correct id", function () {  
    pm.expect(pm.response.text()).to.include("subscriptions/"+environment.subscriptionId+"/resourceGroups/"+environment.rgName+"/providers/Microsoft.ContainerService/ManagedClusters/"+environment.resourceName+"/eventGridFilters/"+environment.egFilterName);  
});
```

GET GetEventGridFilter Comments 0

localhost:8080/subscriptions/subscriptionId/resourceGroups/resourceGroupName/providers/Microsoft.ContainerService/ManagedClusters/resourceName/eventGridFilters/egFilterName

Tests for GET

CORRECT RESPONSE CODE

```
pm.test("Return 200", function () {  
    pm.response.to.have.status(200);  
});
```

CORRECT SUBSCRIPTION RETRIEVED

```
pm.test("Has correct id", function () {  
    pm.expect(pm.response.text()).to.include("subscriptions/"+environment.subscriptionId+"/resourceGroups/"+environment.rgName+"/providers/Microsoft.ContainerService/ManagedClusters/"+environment.resourceName+"/eventGridFilters/"+environment.egFilterName);  
});
```

Postman: Documentation

Motivation

- Create Postman documentation that can be easily followed and tested by other Azure services
- Provide code samples for quick testing of a new API Server
 - Samples provided in 25+ languages and flavors

Example Request GetEventGridFilterTest

```
package main

import (
    "fmt"
    "net/http"
    "io/ioutil"
)

func main() {
```

Example Response 200 OK

Body Headers (3)

```
{
  "id": "subscriptions/subscriptionId/resourceGroups/resourceGroupName/providers/Microsoft.ContainerService/ManagedClusters/resourceName/eventGridFilters/egFilterName",
  "name": "resourceName/egFilterName",
  "type": "Microsoft.ContainerService/ManagedClusters/EventGridFilters",
  "properties": {
    "filter": {
      "subjectBeginsWith": "BeginsWith",
      "subjectEndsWith": "EndsWith",
      "subjectMatches": "Matches"
```

Build Integration and Deployment

Integrating our service within AKS RP

Integrating with AKS Build

- Updating about a dozen Makefiles and other files 😊
- Creating a docker container with eventgrid code
- Documented the steps to make it easier for engineers in the future

Deployment

- Create the EV2 package inside the AKS repo
- Add the deployment pipeline in GitOps
- Add charts for Kubernetes deployment

Next Steps

Finishing the service

Add Publisher Component

- Use Service Bus to detect new versions of Kubernetes that are available
- Send a POST to Event Grid for the right event subscribers

Add MSI Integration

- Use MSI to authenticate for various parts of our service

Finish Deployment

- Create the release pipeline
- Deploy to test region so we can iterate

Our Learnings

Our takeaways from this amazing experience

Software Engineering Principles and Tools

- Creating modular and composable programs
- PR reviews: writing production quality code
- Working with build and deployment
- Software and tools used: Go, Docker, Makefiles, Go modules, cobra command

General

- Working as a remote Software Engineer Intern
- Proper documentation

Thank You!

We'd like to thank Microsoft for giving us this amazing opportunity, especially during these tough times!

A special thanks to:

[Stephane Erbrech](#) for guiding us throughout the internship.

[Marwan Ahmed](#) and [Shoshana Malfatto](#) for mentoring us and answering our many questions.

The AKS Data Plane team for making us a part of their team this summer.

[Ganeshkumar Ashokavardhanan](#)
[Aman Jaiman](#)

Software Engineer Interns

