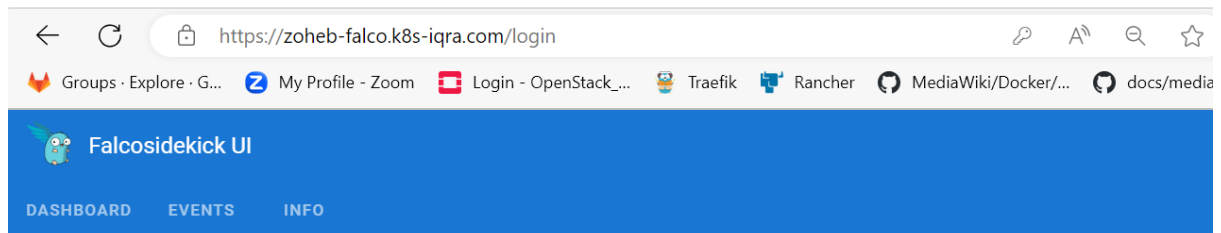


## Scenario:

I've deployed an application in Azure AKS. The pods are up/running, however it's throwing 503 error.

RunWhen Local is also available in the same cluster. Let's troubleshoot:

Access Application: As it's giving 503 error



Login

Password

**LOGIN**

**Error: Request failed with status code 503**

Validate from RunWhen Local command:

```
root@zoh-aks:~# curl -o /dev/null -w '{"http_code": %{http_code}, "time_total": %{time_total}}' -s https://zoheb-falco.k8s-iqra.com  
{"http_code": 503, "time_total": 0.457542}
```

## Find Ingress Owner and Service Health:

The below command is saying it's healthy, but the application is not healthy and throwing 503 error.

```

root@zoh-aks:~# namespace="ingress-nginx"; context="zoh-aks-cilium"; ingress="falco-ingress-external"; echo "Ingress: $ingress"; health_status="NA"; services
=(); backend_services=$(kubectl get ingress "$ingress" -n "$namespace" --context "$context" -ojsonpath='{range .spec.rules[*].http.paths[*]}.backend.service
.name}{ " " }.backend.service.port.number}{ "\n"}(end)'); IFS=$'\n'; for line in $backend_services; do service=$(echo "$line" | cut -d " " -f 1); port=$(echo "
$line" | cut -d " " -f 2); if [ -n "$service" ] && [ -n "$port" ]; then echo "Backend Service: $service, Port: $port"; service_exists=$(kubectl get service "$
service" -n "$namespace" --context "$context" -ojsonpath='{.metadata.name}'); if [ -z "$service_exists" ]; then health_status="Unhealthy"; echo "Validation:
Service $service does not exist"; else endpoint_pods=$(kubectl get endpoints "$service" -n "$namespace" --context "$context" -ojsonpath='{range .subsets[*].
addresses[*]}.Pod Name: {.targetRef.name}, Pod IP: {.ip}(end)'); if [ -z "$endpoint_pods" ]; then health_status="Unhealthy"; echo "Validation: Endpoint fo
r service $service does not have any pods"; else echo "Endpoint Pods"; echo -e "$endpoint_pods"; for pod in $endpoint_pods; do if [[ $pod == *- Pod Name: *
]]; then pod_name=${pod#*- Pod Name: }; pod_name=${pod%%$*}; if [ -n "$pod_name" ]; then owner_kind=$(kubectl get pod "$pod_name" -n "$namespace" --
context "$context" -ojsonpath='{.metadata.ownerReferences[0].kind}'); if [ -n "$owner_kind" ]; then if [ "$owner_kind" = "StatefulSet" ] || [ "$owner_kind"
= "DaemonSet" ]; then owner_info=$(kubectl get pod "$pod_name" -n "$namespace" --context "$context" -ojsonpath='{.metadata.ownerReferences[0].name}') $own
er_kind; else replicaset=$(kubectl get pod "$pod_name" -n "$namespace" --context "$context" -ojsonpath='{.metadata.ownerReferences[0].name}'); if [ -n "$re
plicaset" ]; then owner_kind=$(kubectl get replicaset "$replicaset" -n "$namespace" --context "$context" -ojsonpath='{.metadata.ownerReferences[0].kind}');
owner_name=$(kubectl get replicaset "$replicaset" -n "$namespace" --context "$context" -ojsonpath='{.metadata.ownerReferences[0].name}'); owner_info="$own
er_name $owner_kind"; fi; fi; fi; if [ -n "$owner_info" ]; then echo "Owner: $owner_info"; fi; fi; done; health_status="Healthy"; fi; fi; services+=("$serv
ice"); fi; done; for service in "${services[@]}"; do service_exists=$(kubectl get service "$service" -n "$namespace" --context "$context" -ojsonpath='{.metad
ata.name}'); if [ -z "$service_exists" ]; then health_status="Unhealthy"; echo "Validation: Service $service does not exist"; else endpoint_exists=$(kubectl
get endpoints "$service" -n "$namespace" --context "$context" -ojsonpath='{.metadata.name}'); if [ -z "$endpoint_exists" ]; then health_status="Unhealthy"; e
cho "Validation: Endpoint for service $service does not exist"; fi; fi; done; if [ "$health_status" = "Unhealthy" ]; then echo "Health Status: $health_status
"; echo "====="; elif [ "$health_status" = "Healthy" ]; then echo "Health Status: $health_status"; fi; echo "-----"
Ingress: falco-ingress-external
Backend Service: falco-falcosidekick-ui, Port: 2803
Endpoint Pod:
- Pod Name: falco-falcosidekick-ui-b69f8dcb7-vcfss, Pod IP: 10.0.0.206
- Pod Name: falco-falcosidekick-ui-b69f8dcb7-bgnm4, Pod IP: 10.0.2.13
Owner: falco-falcosidekick-ui Deployment
Health Status: Healthy

```

Ingress: falco-ingress-external

Backend Service: falco-falcosidekick-ui, Port: 2803

Endpoint Pod:

- Pod Name: falco-falcosidekick-ui-b69f8dcb7-vcfss, Pod IP: 10.0.0.206

- Pod Name: falco-falcosidekick-ui-b69f8dcb7-bgnm4, Pod IP: 10.0.2.13

The port used in deployment/service is 2802 and the port used in Ingress backend is 2803.

That's why the traffic is not reaching service, eventually the pod. Resulting 503

Owner: falco-falcosidekick-ui Deployment

Health Status: Healthy

If we can add one more rule to match the deployment/service port with the backend port defined in Ingress. Even the deployment containerport & service port should be same.

If deployment containerport/service port = backend port defined in Ingress, mark it healthy. Otherwise unhealthy.