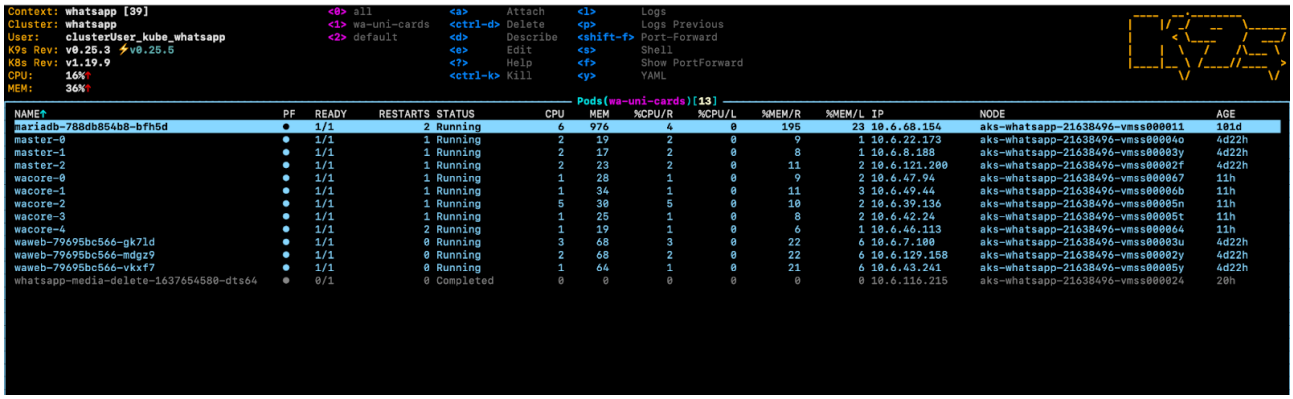
Using Kubernetes in Infra Debugging

Tools to Install

1. Homebrew - Package Manager for macOS (or Linux) **(https://brew.sh/)**
2. Kubectl - The Kubernetes command-line tool, kubectl, allows you to run commands against Kubernetes clusters. You can use kubectl to deploy applications, inspect and manage cluster resources, and view logs. **(https://kubernetes.io/docs/reference/kubectl/)**
3. Kubens - kubens is a tool to switch between Kubernetes namespaces (and configure them for kubectl) easily.
4. Kubectx - kubectx is a tool to switch between contexts (clusters) on kubectl faster. - **(https://developers.redhat.com/blog/2019/05/27/command-line-tools-for-kubernetes-kubectl-stern-kubectx-kubens#)**
5. K9S - K9s is a terminal based UI to interact with your Kubernetes clusters. **(https://k9scli.io/)**
6. Stern - Stern allows you to tail multiple pods on Kubernetes and multiple containers within the pod. Each result is color coded for quicker debugging. **(https://github.com/wercker/stern)**
7. Watch - You can specify the -w or --watch flag to start watching live updates to a particular object.
8. az - To connect to azure kubernetes service

K9S Dashboard

To access the shell of a pod, type k9s in terminal and type ‘s’ when a pod is selected in the dashboard.



On-prem Bots

1. mongo "mongodb://username:password@IP_of_server:port_number"
2. show dbs;
3. use <dbname>;

To find 1st data

db.reports.find({ botId: "", campaignId: ""}).limit(1).pretty();

To find latest data

db.reports.find({ botId: “”, userId: “”}).sort({ created: -1}).limit(1).pretty();

To find controller logs based on secondaryID (stern needs to be installed)

1. kubens services
2. stern -t --since=3h controller | grep "secondaryID/msgID"

To find anything in engagement service,

stern -t --since=1m engagement-service | grep "whatever keyword you want to find"

To export campaign reports from BPCL mongodb

The user should be like “root@mongodb-585db47c7f-j8gz7”

Type this command from inside mongo db pod to export all reports,

mongoexport --uri="mongodb://root:c0mpl1cat3d@10.240.1.253:37017" -d notification -c reports --type=csv -q '{"campaignId" : "apiNotifications"}' --out report.csv --authenticationDatabase admin --fields campaignId,botId,status,userId,currentCronTime,secondaryId,created,comments,_id,

To export only reports of specific dates,

mongoexport --uri="mongodb://root:c0mpl1cat3d@10.240.1.253:37017" -d notification -c reports --type=csv -q '{"created": { "$gte": { "$date": "2022-01-15T00:00:00.000Z"}, "$lt": { "$date": "2022-01-15T23:59:59.999Z" }}, "campaignId" : "apiNotifications"}' --out report.csv --authenticationDatabase admin --fields campaignId,botId,ipAddress,status,userId,currentCronTime,secondaryId,created,customPayload,sentBy,comments,_id

To download exported report to local machine,

Enter this command in regular terminal "issacmathew@Issac-Mathews-MacBook-Air ~ %" ,

kubectl cp dbs/mongodb-585db47c7f-j8gz7:/report.csv ./report.csv

To transfer downloaded report to downloads folder,

cp report.csv ~/Downloads/.

WhatsApp Infra Debugging

1. kubens <namespace>
2. kubectl get pods
3. watch "kubectl get pods" (to stop ctrl+c in Mac)
4. kubectl get configmap db-env -o yaml (To get username and password to enter the DB)
5. kubectl exec -it <mariadb pod name> -- bash
6. mysql -u<username> -p<password>
7. show databases;

Scenario 1 : High Queues

If the response time of a bot for a whatsapp campaign is more than 1.5k ms consistently, then it can be because the bot infra is down/degraded. You can verify this by checking the number of messages present in the callback queue. The number of messages present in the callback queue represents the number of acknowledgement messages sent from Whatsapp Infra to our bot, that have not been processed. Ideally, it should be as close to 0 as possible.

Callback store (Incoming messages):

1. Follow steps (1-7) given above.
2. use <namespace_callbackStore database>
3. show tables;
4. select count(*) from queue;

Jobqueue (Outgoing messages):

5. Follow steps (1-7) given above.
6. use <namespace_jobqueue-Whatsapp database>
7. show tables;
8. select count(*) from queue;

Scenario 2 : Check Image

Recently, we upgraded the docker container images being used for different pods from v2.35 to v2.37.1. Due to this you might face issues for newly created bots. Issues like unable to “Enable/Disable 2 Factor Authentication” or “Send OTP is giving an error in Admin Panel”

1. kubectl get statefulset -o wide
2. kubectl get deploy -o wide

Scenario 3: Whatsapp Infra creation is stuck in admin portal

The admin-backend service is supposed to spawn worker threads to monitor and update the infra status. This process can fail if,

1. If the whatsapp number of the bot does not clear WA validation because the same number is associated with another WA account.
2. The thread times out after 30 mins. If the infra creation takes a lot of time, the thread will exit.
3. The infra is created but the thread fails due to some reason.

If the whatsapp number of the bot does not clear WA validation because the same number is associated with another WA account. In this scenario,

1. kubectx india
2. kubens <namespace>
3. stern -t --since=30m admin-backend
4. Grep or search for logs like **data: { meta: [Object], errors: [Array] }**

If the infra is created but the thread fails due to some reason,

1. Recreate the certificate linked to the bot in the admin portal. This is done so that OTP verification can work. To recreate the certificate, use the API given below:

```
curl --location --request GET 'https://graph.facebook.com/v11.0/187599165530414/phone_numbers?fields=display_phone_number%2Ccertificate%2Cverified_name&access_token='
```

2. Respawn the threads using the API given below:

```
curl --location --request GET 'https://admin.yellowmessenger.com/api/admin/whatsapp/spawn-wa-worker/wa-leap-finance-for-levelup' \ --header 'Cookie: ym_xid=value'
```

3. Use the command `stern -t --since=30m admin-backend` to search for relevant log entries.

4. Signs of successful creation of whatsapp infra are the logs given below:

```
response.status-changeWhatsappPassword 200
isPasswordChanged-NAMESPACE true
```

Scenario 4: High MariaDB CPU usage:

1. We can use `kubectl top pods` command to check current cpu and memory usage.
2. We can check the limit set on mariadb by doing `kubectl get deploy mariadb -o yaml`
3. We can check unders `resources > limits` section and if we find them near to current usage.
4. If we need to increase it to a higher number, we can use `kubectl edit deploy mariadb` and edit the resources section. We can also try to add one more core and 2 more GB memory if we are nearing the limits.

Scenario 5: If Infra cannot be scaled up from UI due to Infra going down:

kubectl get pods | grep -i "Crash" | awk '{print $1}' | xargs -I {} kubectl delete pod {}
This command can be used to scale up from T0 -> T1 and T1->T2

Scenario 6: Media Disk getting full

To check if media disk is getting full run the following command after entering into the namespace,

kubectl exec -it wacore-0 -- df -h

1. One way to fix this would be to manually clear the disk

    (i) Go to the namespace
    (ii) Get inside any one of the wacore pods (all wacore pods share the same media disk)
    (iii) Enter this command to delete files older than 5 days

find /usr/local/wamedia -type f -mtime +5 -exec ls {} \; -exec rm {} \;;

2. Resize disk space if needed

    (i) Go to the namespace and enter this command,

kubectl edit pvc whatsapp-disk

    (ii) Check if disk is resized

kubectl get pvc (if already inside namespace)
kubectl get pvc -n <namespace>

```
nikhilm@Nikhils-MacBook-Pro ~ % export KUBECONFIG=~/ym-r0-whatsapp-k8s.yaml
nikhilm@Nikhils-MacBook-Pro ~ % kubectl get pvc -n wa-sunhera-bharat-9315325433
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mariadb-disk-claim   Bound    pvc-93228344-9968-4e71-916d-aaa5f689127a   25Gi       RWO            default        8d
whatsapp-disk        Bound    pvc-b62089b0-c7e3-4d27-9a96-308544b12ef6   50Gi       RWX            azurefile      8d
nikhilm@Nikhils-MacBook-Pro ~ % kubectl edit pvc -n wa-sunhera-bharat-9315325433 whatsapp-disk
persistentvolumeclaim/whatsapp-disk edited
nikhilm@Nikhils-MacBook-Pro ~ % kubectl get pvc -n wa-sunhera-bharat-9315325433
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mariadb-disk-claim   Bound    pvc-93228344-9968-4e71-916d-aaa5f689127a   25Gi       RWO            default        8d
whatsapp-disk        Bound    pvc-b62089b0-c7e3-4d27-9a96-308544b12ef6   250Gi      RWX            azurefile      8d
nikhilm@Nikhils-MacBook-Pro ~ % kubectl exec -it -n wa-sunhera-bharat-9315325433 wacore-1
error: you must specify at least one command for the container
nikhilm@Nikhils-MacBook-Pro ~ % kubectl exec -it -n wa-sunhera-bharat-9315325433 wacore-1 -- /bin/bash
root@wacore-1:/opt/whatsapp/bin# df -h
Filesystem                                                                                      Size  Used Avail Use% Mounted on
overlay                                                                                         248G  131G  118G  53% /
tmpfs                                                                                            64M     0   64M   0% /dev
tmpfs                                                                                           7.9G     0  7.9G   0% /sys/fs/cgroup
shm                                                                                              64M     0   64M   0% /dev/shm
/dev/sda1                                                                                       248G  131G  118G  53% /etc/hosts
//f5745ec94f82646659219f2.file.core.windows.net/kubernetes-dynamic-pvc-b62089b0-c7e3-4d27-9a96-308544b12ef6  250G   21G  230G   9% /usr/local/wamedia
tmpfs                                                                                           7.9G   12K  7.9G   1% /run/secrets/kubernetes.io/serviceaccount
tmpfs                                                                                           7.9G     0  7.9G   0% /proc/acpi
tmpfs                                                                                           7.9G     0  7.9G   0% /proc/scsi
tmpfs                                                                                           7.9G     0  7.9G   0% /sys/firmware
```