Release migration path

quobis

2nd February 2021

Index

Release 4.2 - Migration path

Quobis communications platform 4.2 is a new step further in order to achieve a complete extensible unified communications framework. New features include better SIP integration with newer available options, better scalability and maintenance through a dedicated API, security features, and multiple updates on the Collaborator applications.

In order to achieve a complete migration from 4.1 to 4.2 you need to achieve the following changes.

Remember that you can submit your questions to support@quobis.com

Migration steps

1. Create a backup of the *Kuberentes* folder, in order to keep the previous configuration.
2. Update the inventory associated with the deployment, including the inventory/group_vars/all/main.yml and inventory/group_vars/all/vault.yml. Default inventory can be taken as reference to update both files.
3. Run the following ansible tags using the sippo-k8s tag described in the next table:
    a. nginx-ingress-redeploy
    b. routing-deploy
    c. erebus-redeploy
    d. oauth2-proxy-redeploy
    e. qss-redeploy
    f. sfu-dispatcher-redeploy
    g. sfu-wrapper-redeploy
    h. sippo-server-redeploy
    i. xmpp-server-redeploy
    j. webphone-redeploy
    k. recording-redeploy (If needed)
    l. sippo-exporter-redeploy
    m. sippo-maintainer-redeploy
    n. kapi-deploy

    Note: The configuration can be changed in the *Kubernetes* folder and the run the previous ansible tag replacing redeploy by restart.

4. Update media server.
    a. audiomixer-deploy
    b. sip-proxy-deploy

Release components

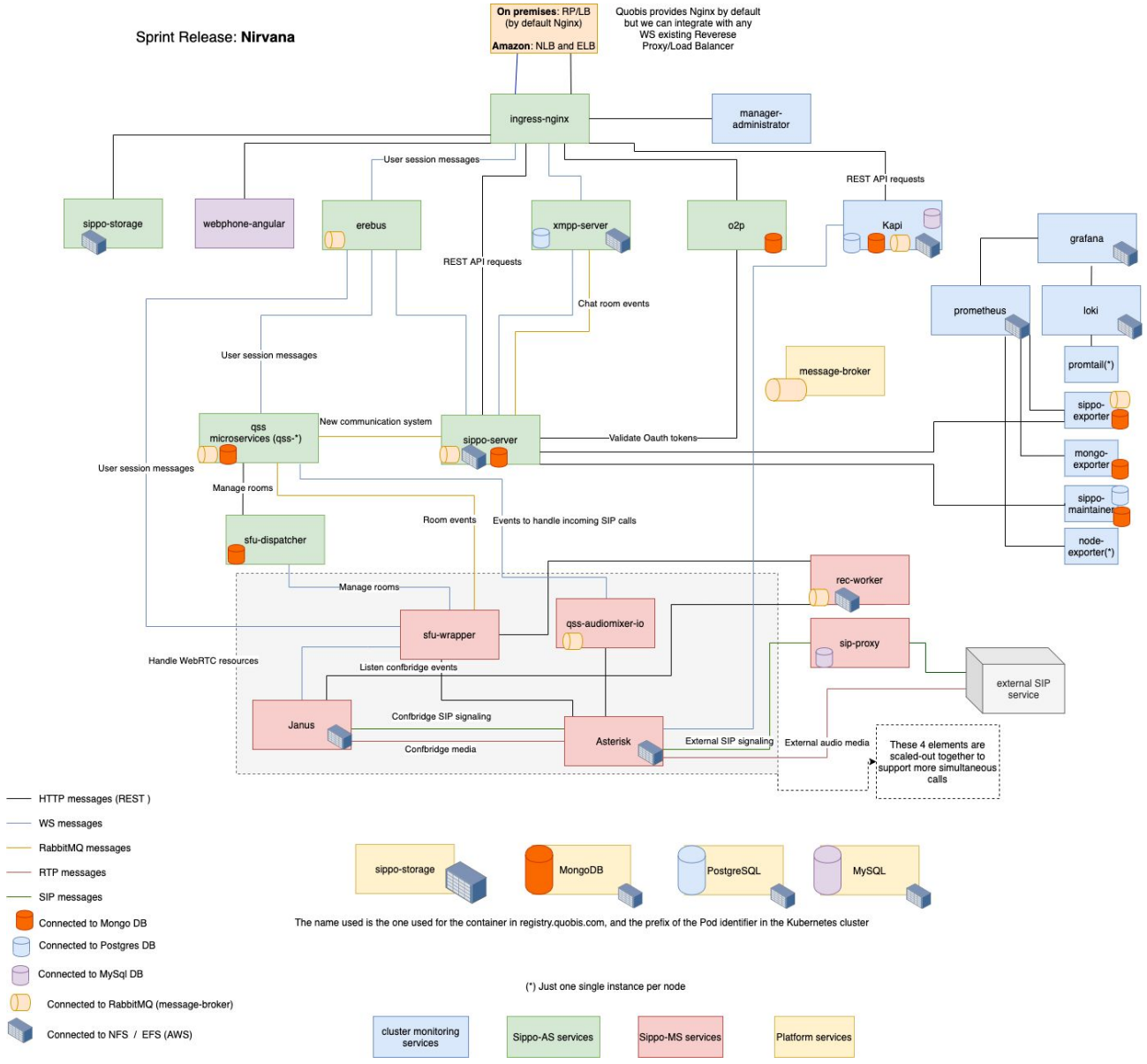
Cluster Element	Version
sippo server	21.1.9
qss	4.12.4
oauth2-proxy	1.15.1
xmpp-server	2.2.3
janus-dispatcher	1.4.3
erebus	1.5.4
janus-wrapper	1.15.6
recording-watchdog	3.2.1
sippoSDK-JS	28.2.1
sippoSDK-Android	0.14.0
sippoSDK-iOS	3.0.0
sippoSDK-cpp	0.1.0
webphone	5.16.6
sippo-maintainer	1.2.6-rc1-kubernetes1.15.3
sippo-exporter	1.5.7
sippo-k8s	2.13.1
kapi	1.4.0-rc2

Note: The previous versions can be subject to changes due to bug fixing.

3rd party software dependencies

3rd party elemene	Version
NodeJS runtime	8.0
message-broker (RabbitMQ)	3.7-management
database (MongoDB)	4.2
reverse-proxy (Nginx)	1.18.0
cluster-ingress (Nginx-ingress)	0.30.0
chat-databsse (PostgresSQL)	12-alpine
audiomixer (Asterisk)	16.7.0
sfu (Janus)	0.9.2
sip-prosy (Kamailio)	5.2.0
sip-proxy-db (MySQL)	5.7
turn-server (CoTURN)	4.5.1.1
monitoring.ui (Grafana)	6.4.3
log-database (Loki)	1.4.0
monitoring-database (Prometheus)	v2.13.0

Architecture changes

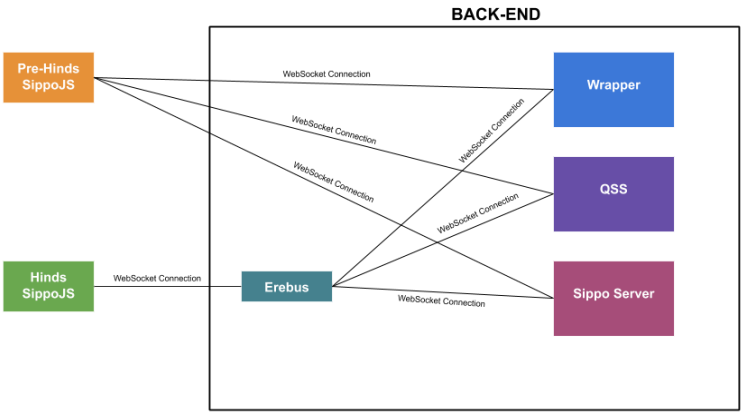


- **New feature: Erebus WebSocket Multiplexing**

This new release includes a new feature affecting the establishment of the connections against the different Sippo services. Now, Erebus includes a WebSockets Multiplexer, making clients able to access all of our services using just a single connection. This change mostly reduces

connection establishment time, which can have a significant impact on call establishment times.

Here is a diagram representing how a pre-Hinds SippoJS will interact with the back-end v how a Hinds SippoJs will do:



Configuration changes

Erebus

erebus

If you are updating or you just want to enable the Erebus WebSocket multiplexer, here is what you have to do:

1. Enable the WebSockets Proxy (a.k.a WebSocket Multiplexer in Erebus). You can do that by adding the following line to the Erebus config/default.toml.

```
[wsproxy]
port = 4000 # Replace 4000 with the port you prefer
```

2. Configure how the WebSockets Proxy will connect to the different services, also in Erebus config/default.toml.

```
# This enables WebSocket multiplexing
[wsproxy]
port = 4000 # Replace 4000 with the port you prefer

# This section configures the connection against the different services
#
# For finding out public URLs you can open a Webphone configured
```

```
# without multiplexing a see open connection
#
# BE CAREFUL SLASHES DO MATTER
#  wss://web.sippo/qss/ and wss://web.sippo/qss are not the same
#
[wsproxy.servers.qss]
server = 'wss://web.sippo/qss/' # This is the public URL where the QSS is exposed
upstream = 'ws://qss-io-ws:8118' # This is the internal URL of the QSS inside the
cluster

[wsproxy.servers.wapi]
server = 'wss://web.sippo/wapi/' # Public Wac Websockets API (WAPI) URL
upstream = 'ws://localhost:3000' # Internal WAPI URL

[wsproxy.servers.wrapperwss]
server = 'wss://wrapper:9033/' # Public secure WebSockets Wrapper URL
upstream = 'ws://wrapper:9032' # Internal Wrapper URL

[wsproxy.servers.wrapperws]
server = 'ws://wrapper:9032/' # Public WebSockets Wrapper URL
upstream = 'ws://wrapper:9032' # Internal Wrapper URL
```

3. Add the WebSockets Proxy URL to the host-meta.json. Most of the time, this will require you to configure your ingress to expose the WebSockets Proxy to the public. You just have to add the section with rel: urn:quobis:sippo:ws-proxy changing the href to the public WebSockets Proxy URL.

```
{
      "links": [
            {
                  "rel": "urn:xmpp:alt-connections:websocket",
                  "href": "wss://web.sippo/xmpp-websocket"
            },
            {
                  "rel": "urn:quobis:xmpp:push",
                  "href": "https://web.sippo/_xpush"
            },
            {
                  "rel": "urn:quobis:fileupload:websocket",
                  "href": "wss://web.sippo/_fsws"
            },
            {
                  "rel": "urn:quobis:sippo:websocket",
                  "href": "wss://web.sippo/wapi/"
            },
            {
                  "rel": "urn:quobis:sippo:https",
                  "href": "https://web.sippo/sapi"
            },
            {
```

```
                    "rel": "urn:quobis:sippo:ws-proxy",
                    "href": "wss://web.sippo/ws/"
                }
        ]
}
```

OAuth2-proxy

oauth2-proxy

All configuration parameters are now on config/default.js

- Configure the ADFS provider in SippoServer to redirect to the O2P instead of the ADFS:

```
{
  id: "adfs",
  ...
  "authorization_endpoint": "https://your.domain.com/o2p/adfs/authorization",
  "authorization_endpoint_mobile": "https://your.domain.com/o2p/adfs/authorization",
  ...
}
```

- Configure the O2P to provide the ADFS resource (default.js):

```
http: {
  ...
  options: {
    ...
    adfs: {
       ....
         resource: 'https://web.sippo.es',
       ...
       authorizationEndpoint:
https://ad.contoso.com/adfs/oauth2/authorize?prompt=login',
       ...
    }
  }
}
```

(!) Note: the default config of O2P is missing the querystring ?prompt=login in authorizationEndpoint.

- Configure the OAUTH_REDIRECT_URI in webphone:

```
OAUTH_REDIRECT_URI: 'https://your.domain.com/webphone/o/adfs/callback',
```

(!) Important: ADFS and O2P should be synchronized (NTP conf)

xmpp-server

xmpp-server

A new module muc_track_occupantshas has been added to the prosody.cfg.lua

Xmpp server will use PostgresSQL for persistency and it has a new MUC occupant presence tracker plugin.

Storage has to be configured in prosody.cfg.lua to use PostgresSQL instead of the internal backend:

```
storage = "sql";
sql = {
       driver = "PostgreSQL";
       database = "xmpp";
       host = "postgresql";
       port = "5432";
       username = "xmpp";
        password = "secret";
}
```

An example can be found here.

If you are upgrading an existing environment you need to migrate the stored data to the new PostGreSQL Database. There is a job migrator to perform this task, you can find the necessary information in Prosody-migrator

Other useful information about PostGre: How to manage postgreSQL

Push notifications are sent via `sippo-server`. To do so, the `push.cfg.lua` file has to be updated to:

```
Component "push.sippo" "push_appserver" http_host = "push.sippo";

push_appserver_external_url = "http://wac:8000/sapi/xmpp/push"
push_notification_with_body = true -- send the message body to remote pubsub node
push_notification_with_sender = true -- send the message sender to remote pubsub node

modules_enabled = {
       "rawdebug";
       "push_appserver_external";
};

modules_disabled = {
       "register";
```

```
};
```

Please adapt hostnames as needed (`http_host` and `push_appserver_external_url`).

WAC

Sippo Server

Two new services are available: supportchat that enables users to request a chat with an agent, and agentstate that exposes the GET agentassign/ API endpoint.

Support Chat and Agent State services require the Agent Assigner. If any of these services are enabled, Agent Assigner MUST be enabled, otherwise the Sippo Server will NOT start

They can be enabled by inserting the following in the`wac.ini` configuration file:

```
; Agent assigner (read the warning above)
[agentassigner]

; Service/SupportChat
;
; Provides all the support chat management features
;
[supportchat]

;
; Service/AgentState
;
; Provides support for agent state
;
[agentstate]
```

With the new version, the concept of AgentSkills has been introduced. This can be used for calling an agent with a particular skill (if we want to call a french speaker agent, for example) but it is also used now for knowing whether a user is an agent or not, causing the deprecation of the capability +c2cagent. Note that the use of the capability, despite beeing considered legacy, it is still available. In order to use the capability, we must set the legacyAgentResolution paramater in agentassigner.toml config file to true, by defautl is false. Note also that the use of the capability does not affect the use of skills for calling a skilled agent, as it only affects to know when a user is an agent.

User capability w3c-contacts-api stop working, now to see the phone contacts in Webphone should be used CONTACTS: LOCAL_SOURCES configuration parameter.

- New param strategy in config/agentassigner.toml that allows to choose between linear or random agent assignation
- Note: legacyAgentResolution must be set to false due to incompatibility with new agentAssigner service.
- Note: [supportchat] must be commented
- Meetings

  Meetings URI prefix is now configurable, previously all meetings had the URL https://my-sippo-server.domain/wac-meeting:meetingId, with this new change the wac-meeting part in the URL is configurable.

  For example a config like this in wac.ini will create meeting URLs with the following format: https://meetings.wac/m/meetingId

```
[meetings]
linkHost = https://meetings.wac/
inviteMechanisms[] = email
inviteMechanisms[] = sms
linkFormat = unique  ; or user
meetingsUri = m/
UserContacts
```

- UserContacts

  UserContacts is a new CRUD service for user contacts. In order to deploy the service, it is necessary to add a line to the wac.ini configuration file, as with any other service, as follow:

```
[usercontacts]
```

- PushNotifications

  The apn_topic parameter no longer has to include the .voip suffix. The service will automatically append it whenever call pushes are sent (and won't do so for chat pushes, respectively).

- UserGroups can now be obtained using the AddressBooks endpoint. To enable this functionality add aggregateGroupContacts to the addressbooks section in the wac.ini:

```
[addressbooks]
```

```
aggregateDomainUsers = true
aggregateUserContacts = true
aggregateUserPhonebooks = true
aggregateStaticContacts = true
aggregateGroupContacts = true
```

Note: it's important to apply the correct contacts configuration for the client to avoid possible contact visualization problems.

- Backend checks `fileUpload` service has been added. Service configuration on `wac.ini` should be updated.
    - Source IP, allowed extensions, file size and filename size could be limited.
    - If not configuration options provided by default, protection is applied:
        - extensions as published on public documentation, 40 characters max and 40MB as max file size.

```
[fileupload]
;
address = 0.0.0.0
allowedExtensions[] = image/jpeg
allowedExtensions[] = image/png
allowedExtensions[] = video/mp4
allowedExtensions[] = audio/mp4
allowedExtensions[] = text/plain
maxFileSize = 4194304
maxFilenameLength = 40
```

- Now meetings `remindBefore` parameter will be used for creating meeting alarms.
- The templates for the email that a meeting owner will receive, must be added in a folder called owner inside the folder for particular language. For instance, if the templates are under `config/meetings/tpl/email/[LANG]/TEMPLATE_FILES`, the templates for the meeting owner must be at `config/meetings/tpl/email/[LANG]/owner/TEMPLATE_FILES`
- Add service `[pushnotificationsinfo]` to properly build push notification data in QSS
- Removal of devices contact backend
- Add alerts service if desired to wac.ini as the others services.
- Password policy is now configurable by using several parameters. This can be configured at config/passwordpolicies.json
    - The rules that can be defined must be of the following types:
        - must, password must be compliant with every rule in the rules array

        - some, password must be compliant with at least count rules in the rules array. (a some with count = rules.length is a must, a some with count = 0 is a true)
    - When more than one rule is defined the password MUST be compliant with all of them.
    - The rules can be of the following type:
        - digits, the password must contain at least the specified number of digits (when null defaults to 1)
        - letters, the password must contain at least the specified number of letters (when null defaults to 1)
        - lowercase, the password must contain at least the specified number of lowercase letters (when null defaults to 1)
        - uppercase, the password must contain at least the specified number of lowercase letters (when null defaults to 1)
        - symbols, the password must contain at least the specified number of symbols (when null defaults to 1)
        - spaces, the password must contain at least the specified number of spaces (when null defaults to 1)
        - min, the password must contain at least that amount of characters (not nullable)
        - max, the password must contain at most that amount of characters (not nullable)
        - oneOf, the passwords must be one of the passwords in the list (not nullable)
        - The rules: digits, letters, lowercase, uppercase, symbols and spaces can be inverted by putting them in a { not: rule } object. i.e: if we want no digits in the password: { not: { digits: null } }
- Password policy configuration example:

```
{
      "passwordPolicies": [
            {
                  "type": "must",
                  "rules": [{"not": {"spaces": null}}, {"min": 9}, {"max":
100}]
            },
            {
                  "type": "some",
                  "count": 3,
                  "rules": [
                        {"digits": 1},
                        {"lowercase": 1},
                        {"uppercase": 1},
                        {"symbols": 1}
                  ]
            }
      ]
}
```

COLLABORATOR-WEB

WEBPHONE

- Deprecation of the following CONF parameters:

```
CONFERENCE_LOG = {
        DATE_GROUPS: true,
        DATE_GROUP_FORMAT: 'dd/MM/yyyy',
        DURATION_FORMAT: 'mm:ss',
        SHOW_TODAY: (true as boolean | string),
        SHOW_YESTERDAY: (true as boolean | string),
    };
```

- New CONF parameters:

```
    /**
     * Object that allows to change the date and duration format
     *
     * - DATE[string]: date format for example 'dd/MM/yyyy'
     *
     * - DURATION_FORMAT[string]: duration format for example 'mm:ss'
```

```
 */
DATE = {
    DATE_FORMAT: 'dd/MM/yyyy',
    DURATION_FORMAT: 'mm:ss',
};
```

- The `OAUTH_REDIRECT_URI` and `OAUTH_INTERCEPT_URI` parameters are not needed now in the `CONF.js` file. All configuration files of the project are going to be updated in this release.
- Updated how to customize the new webphone color palette with Jenkins. -> https://quobis.atlassian.net/wiki/spaces/DevTeam/pages/2015559682/How+to+customize+Webphone+colors+from+jenkins
- Include the following parameter in CONF.js file: MEETINGS_PATH: 'm'

```
MEETINGS_PATH: 'm',
```

DISPATCHER

SFU-DISPATCHER

- Two new parameters added in quobis-dispatcher-config.js:
    - secret: used to generate dynamic turn credentials, this secret should be the same configured in coturn. If the secret is not specified, the old system will be used.
    - expirationHours: specify the valid time for credentials.
- Now config will be placed in /opt/sippo/dispatcher/lib/quobis-dispatcher-config.js

QSS

QSS

- The enforceRecording param has been removed
- Public rooms now have an specific URL schema defined by quickConference service.
- Public rooms allow our users to quickly create conference rooms for non scheduled casual talks. Meetings on the other hand are scheduled pre-created events that have a start and finish dates. In this release, these concepts diverge, now the "Quick Conference" service allows to customize endpoint for public rooms.
- If you want to enable the Quick Conference service (you should, otherwise your deployment will lose functionality) you must start the Quick Conference service in the

QSS and configure the Quick Conference URL in the QSS config. The service can be launched like any other QSS service, it is called quickConference. The following lines should be added to the QSS config:

```
      "quickConference": {
            "uriRegex": "^wac-conf:[\\d\\w]{3,}$"
      },
```

- A new `Calls` service must be deployed. It can be launched like any other QSS service. The service name is `calls`.
- Change audiomixer hostname in `audiomixerio-config-sfu1.json` file, `audiomixer-sfu1-0` instead of `hn-am-sfu1`:

```
{
  "service": {
    ...
    "audiomixersio": {
      "asterisk": {
        ...

"mixers":["audiomixer-sfu1-0.audiomixer-sfu1.default.svc.cluster.local:5038"]
      }
    },
  ...
  }
}
```

Change additional `audiomixerio-config-sfu2.json` file too if it exists.

WRAPPER

JANUS-WRAPPER

- A new config parameter recordingType has been added at quobis-janus-config.js under config.janus. This parameter allows to make recordings of only video (recordingType: video), only audio (recordingType: audio) or both (recordingType: all , default one) and to disable the recording (recordingType: none).

Change audiomixer hostname format from `hn-am-sfu1` to `audiomixer-sfu1-0` in `quobis-janus-config-sfu1.js`:

```
config.asterisk = {
        sip: "audiomixer-sfu1-0.audiomixer-sfu1.default.svc.cluster.local:5080",
        ...
        ami: {
                ip: "audiomixer-sfu1-0.audiomixer-sfu1.default.svc.cluster.local",
                ...
        }
};
```

Change additional `quobis-janus-config-sfu2.js` file too if it exists.

Audiomixer

Asterisk

- SIP on hold

  Add the next context to extensions.conf file:

  ```
  [hold]
  exten => _X.,1,NoOp("Call on hold context, go to Stasis application")
  same => n,Set(PJSIP_MOH_PASSTHROUGH()=yes)
  same => n,Stasis(call-on-hold)
  same => n,NoOp("Return from Stasis application")
  ```

  Add the next load module actions to modules.conf file:

  ```
  load = app_stasis.so
  load = res_stasis.so
  load = res_stasis_answer.so
  load = res_stasis_recording.so
  load = res_stasis_playback.so
  load = res_stasis_snoop.so
  load = res_http_websocket.so
  load = res_ari.so
  load = res_ari_model.so
  load = res_ari_channels.so
  load = res_ari_applications.so
  load = res_ari_events.so
  ```

Change sfu and sip-proxy hostname format from `hn-<name>` to `<name>-0` in `pjsip-audiomixer-sfu1.conf` as follows:

```
[janus-identify]
...
match=sfu1-0.sfu1.default.svc.cluster.local

[kamailio]
...
contact=sip:sip-proxy-1-0.sip-proxy-1.default.svc.cluster.local
```

Change `pjsip-audiomixer-sfu2.conf` file too if it exists.

- Include file http.conf in the directory kubernetes/audiomixer/configmaps/ when you need to include the media server inside the cluster.

KAPI

KAPI

- SIP on hold

  In order to activate the “SIP on hold” feature you need to activate and manage in the kapi deployment the next params:

  ```
        - name: EXTENDED_ONHOLD
          value: "false"
        - name: SECONDS_TO_BYE
          value: "3"
  ```

  If EXTENDED_ONHOLD is true a BYE will be sent after SECONDS_TO_BYE seconds when you invoke /sapi/kmanage/siponhold endpoint.

- KAPI documentation

  Nginx ingress - Add the next routing to the ingress in order to serve the documentation URL of the KAPI in your deployment. You need to add a new file kapi.yml, change the <serverURL> to the correct one in your deployment and activate from ansible the variable enableKapiui:

  ```
  apiVersion: extensions/v1beta1
  kind: Ingress
  metadata:
    name: kapi
  ```

```
  namespace: default
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/enable-cors: "true"
    nginx.ingress.kubernetes.io/cors-allow-methods: "PUT, GET, POST, OPTIONS,
DELETE"
    nginx.ingress.kubernetes.io/cors-allow-headers: "Content-Type,
Authorization, Content-Length, X-Requested-With"
    nginx.ingress.kubernetes.io/cors-max-age: "86400"
    nginx.ingress.kubernetes.io/cors-allow-credentials: "true"
    nginx.ingress.kubernetes.io/configuration-snippet: |
      more_set_headers "Pragma: no-cache";
      more_set_headers "Cache-Control: no-cache, no-store";
      more_set_headers "X-XSS-Protection: 1, mode=block";
      more_set_headers "X-Frame-Options: SAMEORIGIN";
      more_set_headers "X-Content-Type-Options: nosniff";
    nginx.ingress.kubernetes.io/cors-allow-origin: "https://<serverURL>"
    nginx.ingress.kubernetes.io/rewrite-target: /$1
spec:
  rules:
  - host: <serverURL>
    http:
      paths:
      - path: /sapi/kmanage/ui/(.*)
        backend:
          serviceName: kapi
          servicePort: ui
```

Modify sapi.yml adding the next lines:

```
- path: /sapi/kmanage
    backend:
     serviceName: kapi
     servicePort: insecure
```

SIP-proxy

Kamailio

Change sip-proxy hostname format from `hn-sip-proxy-1` to `sip-proxy-1-0` and `audiomixer-sfu1-0` instead of `hn-am-sfu1`, in `kamailio-sip-proxy-1.conf`:

```
route[MANIPTOASTERISK] {
                ...
#!ifdef WITH_REGISTRATION
                ...
                $rd = "audiomixer-sfu1-0.audiomixer-sfu1.default.svc.cluster.local";
                ...
```

```
#!endif
                ...
}

route[FROMASTERISK] {
                ...
                record_route_preset("10.1.21.44:5060;r2=on",
"sip-proxy-1-0.sip-proxy-1.default.svc.cluster.local:5060;r2=on");
                ...
}

route[TOASTERISK] {
                ...

record_route_preset("sip-proxy-1-0.sip-proxy-1.default.svc.cluster.local:5060;r2=on",
"10.1.21.44:5060;r2=on");
                ...
}
```

Change `kamailio-sip-proxy-2.conf` file too if it exists.

To use `sipAuthentication` feature, include the following changes in the `kamailio.cfg` file, the indicated numbers are using the `kamailio.cfg.j2` template of the previous version as reference.

Enable the directives, line 4:

```
#!define WITH_AUTHENTICATION
#!define WITH_DIALOG
#!define WITH_SQLOPS
```

Load the modules, line 336:

```
#!ifdef WITH_DIALOG
loadmodule "dialog.so"
#!endif

#!ifdef WITH_SQLOPS
loadmodule "sqlops.so"
#!endif
```

Set module parameters, line 521:

```
#!ifdef WITH_DIALOG
#!ifdef WITH_AUTHENTICATION
modparam("dialog", "track_cseq_updates", 1)
```

```
#!endif
#!endif

#!ifdef WITH_SQLOPS
modparam("sqlops","sqlcon","ca=>mysql://kamailio:{{ kamailio_ps
}}@sip-proxy-database.{{ namespace }}.svc.cluster.local/kamailio")
#!endif

#!ifdef WITH_AUTHENTICATION
modparam("uac", "auth_username_avp", "$avp(auser)")
modparam("uac", "auth_password_avp", "$avp(apass)")
modparam("uac", "auth_realm_avp", "$avp(arealm)")
#!endif
```

Insert the code in the request_route block, line 1048, route[MANIPFROMASTERISK] and “REFER“ condition:

```
#!ifdef WITH_AUTHENTICATION
                        t_on_failure("SIPAUTH");
#!endif
```

Insert the code in the request_route block, line 1052, route[MANIPFROMASTERISK]:

```
#!ifdef WITH_AUTHENTICATION
                if (has_totag() && (is_direction("downstream"))) {
                        $var(a) = $cs;
                        $var(b) = $(var(a){s.int});
                        $var(c) = $var(b) + 1;
                        remove_hf("Cseq");
                        append_hf("Cseq: $var(c) $rm\r\n");
                }
#!endif
```

Insert the code in the request_route block, line 1118, route[FROMASTERISK]:

```
#!ifdef WITH_AUTHENTICATION
                        if (!has_totag())
                                dlg_manage();
                        t_on_failure("SIPAUTH");
#!endif
```

Create a new failure_route block, at the end of kamailio.cfg file:

```
# Manage authentication routing cases
failure_route[SIPAUTH] {
```

```
        route(NATMANAGE);
        if (t_is_canceled()) exit;
#!ifdef WITH_AUTHENTICATION
        if (t_is_canceled()) {
                exit;
        }
        if (t_check_status("401|407")) {
                xlog("L_INFO", "Received 401 or 407 and $fU user\n");

                sql_query("ca", "select auth_username,auth_password from uacreg where
l_username='$fU';", "ra");
                if ($dbr(ra=>rows)>0) {
                        xlog("L_INFO", "Authenticating $fU user for $rm request\n");
                        $avp(auser) = $dbr(ra=>[0,0]);
                        $avp(apass) = $dbr(ra=>[0,1]);
                }
                     sql_result_free("ra");
                uac_auth();
                t_relay();
                exit;
        }
#!endif
}
```

- In addition, it is necessary to deploy the following items:
    - Create mysql_ps and kamailio_ps passwords as usual.
    - Create EFS/NFS volumes to persistent data if it does not exist.
    - Deploy sip-proxy-database.

CoTURN

turn-server

Ability to configure the system to use dynamic TURN credentials to prevent fraudulent usage of the media server and DoS attacks. This functionality is configured by setting two new parameters in file quobis-dispatcher-config.js (“secret” and “expirationHours”)

Additionally, the following settings must be set in the coTURN server configuration:

```
use-auth-secret
static-auth-secret=<credential-secret>
```

Being <credential-secret> the same one used in the Dispatcher configuration.