Synapse Section: Additional Config

The `Additional Config` section, which allows including config not currently configurable via the UI from the [Configuration Manual](#), is available under the 'Advanced' section of the `Synapse` page.

> We strongly advise against including any config not configurable via the UI as it will most likely interfere with settings automatically computed by the updater. Additional configuration options are not supported so we encourage you to first raise your requirements to Support where we can best advise on them.

Additional

Edited

Additional config to inject

```
 1   retention:
 2     enabled: true
 3     default_policy:
 4       min_lifetime: 1m
 5       max_lifetime: 5m
 6     allowed_lifetime_min: 1m
 7     allowed_lifetime_max: 5m
 8     purge_jobs:
 9       - interval: 3m
10   server_notices:
11     system_mxid_localpart: notices
12     system_mxid_display_name: "Notices"
13     room_name: "Notices"
14   auto_join_rooms:
15     - "#apple:example.com"
16     - "#banana:example.com"
17     - "#orange:example.com"
18   autocreate_auto_join_rooms: true
19   auto_join_mxid_localpart: system
```
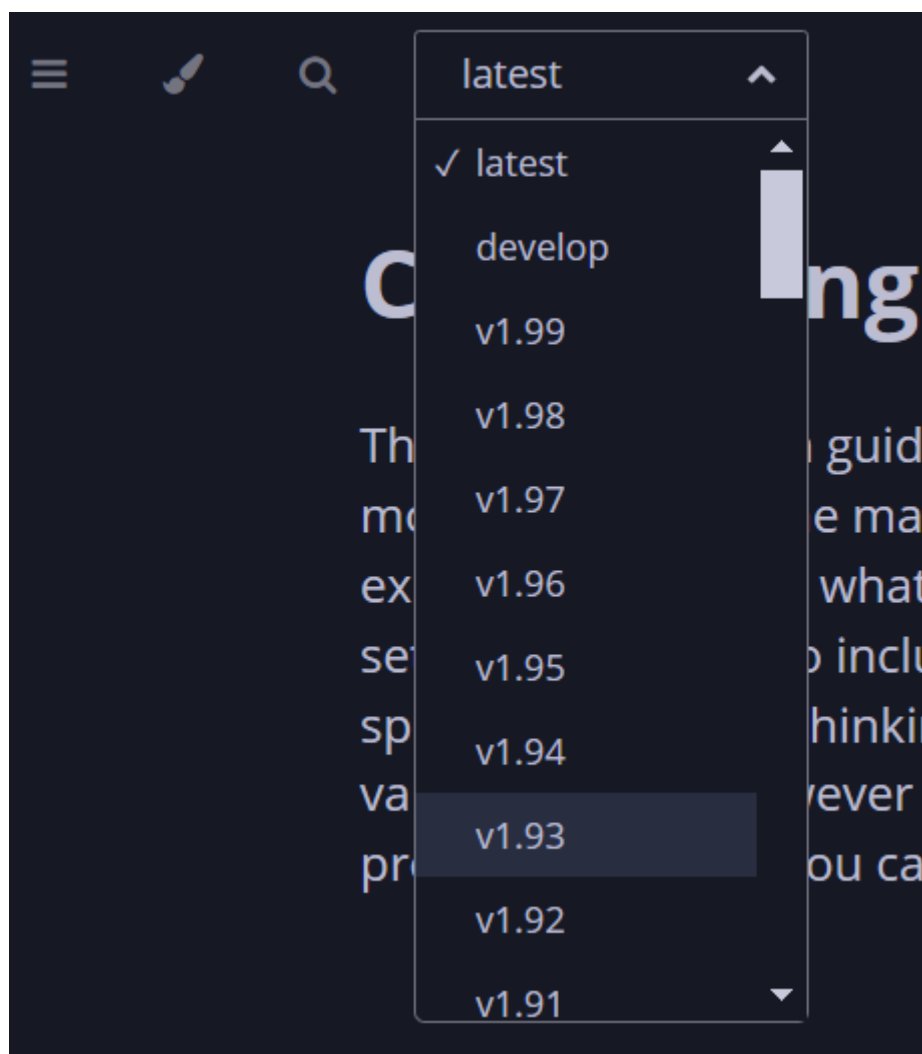
Configuration should follow the same format as supplied by the [Configuration Manual](#), if you include options that have otherwise been configured via the UI they will be overridden with the exception of MAU, Federation and

Data Retention (see Nonoverridable Config). Though as noted above, any additional config carries the risk that it will most likely interfere with settings automatically computed by the updater.

What version of Synapse am I running?

> Remember to set the configuration manual page to the version of Synapse deployed by the installer, otherwise you may see configuration options / guidance not applicable to the version of Synapse you have deployed.

You can determine the version of Synapse you have deployed by using `kubectl describe pod first-element-deployment-synapse-main-0 -n element-onprem | grep version`, changing the pod name as needed. This will output something like `app.kubernetes.io/version=v1.93.0-lts.1-base`, as such when you visit any link to the Configuration Manual, you should update the page to see the correct information for your version.

Known Issues

`max_mau_value`, `limit_usage_by_mau`, `federation` and `retention`

Configuration of these via `Additional Config`, that are in conflict with those set via the UI, will not override the UI set values. As such, we do not advise including them or any related settings within the `Additional Config` as they are of increased risk to causing issues with your deployment.

`auto_join_rooms`

Due to how the installer sets up Synapse, the `auto_join_rooms` option will only work when configured as required on the first deployment. Should you configure this on an existing deployment, or change the rooms on a subsequent deployment, it will not function and you'll receive various errors within the Synapse pod logs. To resolve you will need to manually create the rooms and specify `auto_join_mxid_localpart` in your config. If you're using AdminBot / AuditBot, either would be a perfect candidate for the specified MXID as you can be sure they will be in any room you specify.

Therefore in order to get this setup, you'll need to follow these steps:

- For a brand new "fresh" install, simply specify with config per the manual, on the first user registration, they will create and join the specified rooms and all subsequent users will also auto-join.

  ```
  auto_join_rooms:
    - "#exampleroom:example.com"
    - "#anotherexampleroom:example.com"
  ```

- For existing installs, or when you wish to adjust the auto-join room list, you will need to:
  1. Manually create the rooms and assign the desired alias. (Room Settings -> Local Addresses)
  2. Add the following config, making sure to set the localpart to a user present within the rooms specified. This could be the room creator, someone invited who has joined, or something like Admin/Audit Bot.

     ```
     auto_join_mxid_localpart: adminbot
     ```

  3. Redeploy, wait for the synapse pod to restart
  4. Newly registered users will now auto-join the specified rooms

As usual, with `auto_join_rooms`, the caveat is that changing the rooms will not automatically join previously registered users to the updated rooms. To automate this you will likely need to make use of the Admin API, see Using Python with the Admin + Client-Server APIs, specifically Example #1: Join Users to Rooms would be a good starting point.

Exceptions

While use of Additional Config is not recommended, there are certain circumstances built-in to the UI that will allow you to defer to configuration options you will need to specify within the Additional Config block. These exceptions will be covered here, however please be advised, using them still carries risk of instability so we'd recommend sticking with options fully supported by the UI itself.

`Custom` Registration

Within the Synapse section of the installer, as part of the registration configuration, you can select `Custom`. When doing so, configuration of [Registration](#) should be done via Additional Config, allowing you more control.

Options that can be configured can be found at the linked [Registration](#) section of the Synapse Configuration Manual, but include:

- [enable_registration](#)
- [enable_registration_without_verification](#)
- [registrations_require_3pid](#)
- [registration_requires_token](#)
- [registration_shared_secret](#)

Allowing Private Federation via [ip_range_whitelist](#)

By default private IP ranges are blacklisted, per `ip_range_blacklist`. So when looking to privately federate between two homeservers, where they'd communicate over one of these private ranges, without specifying said range using `ip_range_whitelist` it will fail showing errors like the below:

```
synapse.http.federation.well_known_resolver - 259 - INFO - GET-369 - Fetching
https://server2.example.com/.well-known/matrix/server
synapse.http.client - 199 - INFO - sentinel - Blocked 172.20.8.127 from DNS resolution to server2.example.com
```

To resolve this, you will need to add the following to the Additional config:

```
ip_range_whitelist:
   - '172.16.0.0/12'
```

Config Example

When setting additional config via the UI, the following would be added to the your `deployment.yml`:

Additional

Edited

Additional config to inject

```
1   ip_range_whitelist:
2     - '172.16.0.0/12'
```

```yaml
spec:
 components:
  synapse:
   config:
    additional: |-
      ip_range_whitelist:
        - '172.16.0.0/12'
```

Revision #18
Created 8 February 2024 14:05:30 by Kieran Mitchell Lane
Updated 24 May 2024 10:29:25 by Kieran Mitchell Lane