

Приемы парсинга событий

Парсинг нестандартной даты

Дата Время

#D#

220523;080050;5522.5957;N;08605.0047;E;0;65;142.400000;22;NA;NA;NA;NA;NA;fig:1:7,mileage:2:1097.000000,timp:1:0,pwr_ext:2:13.700000,battery:1:41,sum_acc:2:1.140000,amtr_x:1:2,wln_accel_max:2:0.020000,wln_brk_max:2:0.000000,amtr_y:1:0,wln_crn_max:2:0.000000,amtr_z:1:0,hours_koef:1:16777216,tls1:1:0,cls1:1:0,fls1:1:3,mid:1:1,cmd:1:149

Нужно привести к такому виду:

ISO8601 (Syslog timestamp)

`[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}(\.[0-9]{+})?([zZ]|([+-])([01]\d|2[0-3])):?(([0-5]\d)?)`

2020-03-12T13:34:56.123Z INFO [org.example.Class]: This is a #simple #logline containing a 'value'.

Как выглядит парсер:

Условия дополнительной нормализации

Схема нормализации

Обогащение

*Название

D

*Метод парсинга

csv

?

*Разделитель

;

*Сохранить дополнительные поля

Да

Примеры событий

Сопоставление

Исходные данные	Поле KUMA	Подпись	
0	FlexString1		
1	FlexString1Label		

* Тип преобразования

replaceWithRegex

✕

Редактирование

(\d\d)(\d\

\$3-\$2-\$1

* Тип преобразования

replaceWithRegex

✕

Редактирование

(\d\d)(\d\

\$1:\$2:\$3

Далее работаем склеиваем эти поля шаблоном и обнуляем

Дополнительный парсинг событий

Ветвление событий от beats в зависимости от input типа

Даны следующие типы событий (содержимое тестового сообщения сокращено для лучшего понимания):

```
{ "tags": ["beats_input_raw_event"], "input": { "type": "filestream" } }  
{ "message": "I0130 14:38:47.090079 1837403 utils.go:187] ID: 544472 GRPC response:"  
{ }, "input": { "type": "container" } }  
{ "journal": { "system": "true" }, "tags": ["beats_input_codec_plain_applied"], "input": { "type": "journald" } }  
{ "input": { "type": "journald" }, "journal": { "system": "true" }, "tags": ["beats_input_codec_plain_applied"] }  
{ "journal": { "system": "true" }, "input": { "type": "journald" }, "tags": ["beats_input_codec_plain_applied"] }
```

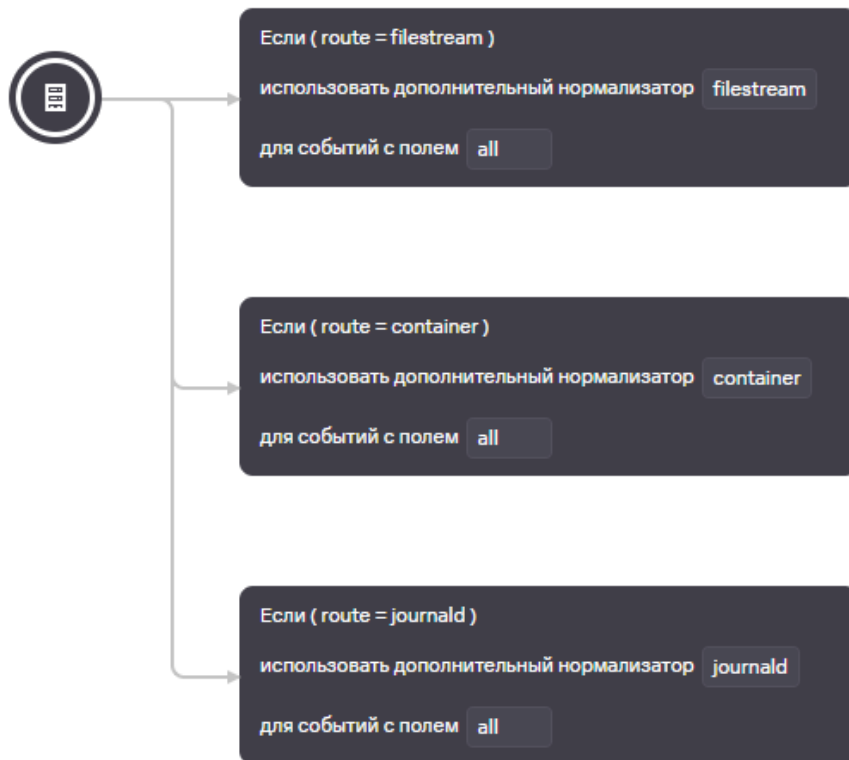
Необходимо в парсинге разветлять (тк у каждого типа свой набор полей) парсинг в зависимости от типа input поля, мы имеем три типа в данном примере:

- "input": { "type": "container" }
- "input": { "type": "journald" }
- "input": { "type": "filestream" }

Причем, поле input может находиться как в начале, так и в середине, и в конце сообщения. Поэтому для ветвления в первом шаге парсинга будут использоваться регулярные выражения:

*Метод парсинга	<div>regex</div> <div>?</div>
*Сохранить исходное событие	<div>Всегда</div>
*Сохранить дополнительные поля	<div>Да</div>
Описание	<div>Описание</div>
Примеры событий	
*Нормализация	<div><input type="checkbox"/> Использовать синтаксис CEF при нормализации</div>
	<div><div>⋮</div><div><div>{?P<all>.*?"input":{\ "type"\:\\"(?P<route>filestream)\\"}.*}</div><div>×</div></div></div>
	<div><div>⋮</div><div><div>{?P<all>.*?"input":{\ "type"\:\\"(?P<route>journald)\\"}.*}</div><div>×</div></div></div>
	<div><div>⋮</div><div><div>{?P<all>.*?"input":{\ "type"\:\\"(?P<route>container)\\"}.*}</div><div>×</div></div></div>

Поле из regex с наименованием route будет использоваться для маршрутизации по условию в нужный парсер, поле all необходимо для передачи полного содержимого в подпарсер. Структура парсера выглядит следующим образом:



Рассмотрим один подпарсер, например, filestream:

Дополнительный парсинг событий

Условия дополнительной нормализации

Схема нормализации

Обогащение

Поле, которое следует передать в нормализатор:

all

И ▾

+ Добавить условие

+ Добавить группу

route

= ▾

filestream

✕

Тк общая структура сообщения формата JSON, используется соответствующий коробочный парсер:

*Название

filestream

*Метод парсинга

json



*Сохранить дополнительные поля

Да

Парсинг массивов

Актуально для KUMA 3.0+

В KUMA 3.0.2 появилась возможность создания кастомных полей типа "массив" (SA, NA, FA), доступные для методов парсинг JSON и KV. Чтобы записать массив в дополнительное поле, достаточно его указать в маппинге:

Сопоставление

+ Добавить строку

Удалить



Исходные данные

Поле KUMA



commandLine



SA.commandLine



В событии это будет выглядеть следующим образом:

Extension fields

SA.commandLine

netstat,-t,-l

Если с массивом в таком случае работать не удобно и нужно все элементы из массива "склеить" через delimiter и записать в отдельное поле, можно воспользоваться обогащением. Для этого сначала массив мапится на строковое поле:

Тип источника данных*	событие
Исходное поле*	SA.commandLine
Целевое поле*	DeviceCustomString1

В таком случае в событии данное поле будет представлять собой массив переведенный в строку:

DeviceCustomString1 ['netstat','-t','-l']

Чтобы привести ее в более "приятный" вид можно выполнить следующие преобразования:

Тип источника данных*	событие
Исходное поле*	DeviceCustomString1
Целевое поле*	DeviceCustomString1
Отладка	<input type="checkbox"/>
Преобразование 1	
Тип*	replaceWithRegexp
Выражение	^\\[
Чем заменить	чем заменить
Преобразование 2	
Тип*	replaceWithRegexp
Выражение	\\\$
Чем заменить	чем заменить
Преобразование 3	
Тип*	replace
Символы	::
Чем заменить	

После этого в DeviceCustomString1 будут записаны все элементы массива через выбранный в последнем (3) преобразовании делитель (в данном примере это "пробел"):

DeviceCustomString1

netstat -t -l

Передача сырого события в экстранормализатор, для доступа к элементам массива

Актуально для KUMA 3.0+

Для передачи «сырого» события в экстра-нормализатор необходимо:

- открыть нормализатор событий;
- перейти в меню «Условия дополнительной нормализации»;
- активировать параметр «Использовать сырое событие».

По умолчанию параметр «Использовать сырое событие» не активен.

Extra normalization conditions

Normalization scheme

Enrichment

Use raw event*

Yes

Filter parameters

⊕

AND + Add condition + Add group

Event.System.EventID	⌵⌵⌵	=	▼	216	x
Event.System.Provider.Name	⌵⌵⌵	=	▼	ESENT	x

Рекомендуется активировать параметр «Использовать сырое событие» в нормализаторах типа «xml», «json».

Для передачи «сырого» события в экстра-нормализатор второго, третьего и более глубоких уровней вложенности необходимо последовательно включить параметра «Использовать сырое событие» в каждом экстра-нормализаторе по пути следования события в целевой экстра-нормализатор и непосредственно в целевом экстра-нормализаторе.

В качестве примера работы данной функции вы можете обратиться к нормализатору Microsoft Products для KUMA 3.0.1: параметр «Использовать сырое событие» включен последовательно в экстра-нормализаторах «AD FS» и «424».

В качестве примера, событие:

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider
Name='ESENT'><EventID
Qualifiers='0'>216</EventID><Level>4</Level><Task>3</Task><Keywords>0x80000000000000
0</Keywords><TimeCreated SystemTime='2024-01-
20T20:06:07.144730300Z'><EventRecordID>870234</EventRecordID><Channel>Application</C
hannel><Computer>COMPANY.COM</Computer><Security/></System><EventData><Data>Isa
ss</Data><Data>724,R,98</Data><Data></Data><Data>C:\Windows\NTDS\ntds.dit</Data><D
ata>\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy50\Windows\NTDS\ntds.dit</Data></Ev
entData></Event>
```

При парсинге ID события 216:

Extra normalization conditions **Normalization scheme** Enrichment

Name*

ESENT_216

Parsing method* ⓘ

xml

XML attributes

Tag numbering

Event.EventData.Data x

Keep extra fields*

Yes

Event examples

Mapping

+ Add row

🗑 Delete

<input type="checkbox"/>	Source		KUMA field	Label	Examples
<input type="checkbox"/>	Event.EventData.Data.0	⇅⇅	SourceProcessName	▼	
<input type="checkbox"/>	Event.EventData.Data.3	⇅⇅	OldFilePath	▼	
<input type="checkbox"/>	Event.EventData.Data.4	⇅⇅	FilePath	▼	

Будет корректно разбираться:

SourceProcessName	Isass
Service	BorisTest(tcp/5577)
ExternalID	1276974
FilePath	\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy47\Windows\NTDS\ntds.dit
OldFilePath	C:\Windows\NTDS\ntds.dit
Type	Base

Смена порядка следования экстранормализаторов

Материал был предоставлен пользователем комьюнити ♥

По умолчанию в GUI KUMA отсутствует возможность перемещать экстранормализаторы внутри правила нормализации и менять их местами. Однако, в ряде случаев данная операция всё же требуется. Например, когда нужно добавить блок с экстранормализатором выше уже существующих, так как они проверяются последовательно. Через веб-интерфейс это сделать проблематично, т.к. потребуются удаление и пересоздание заново всех блоков экстранормализаторов идущих ниже.

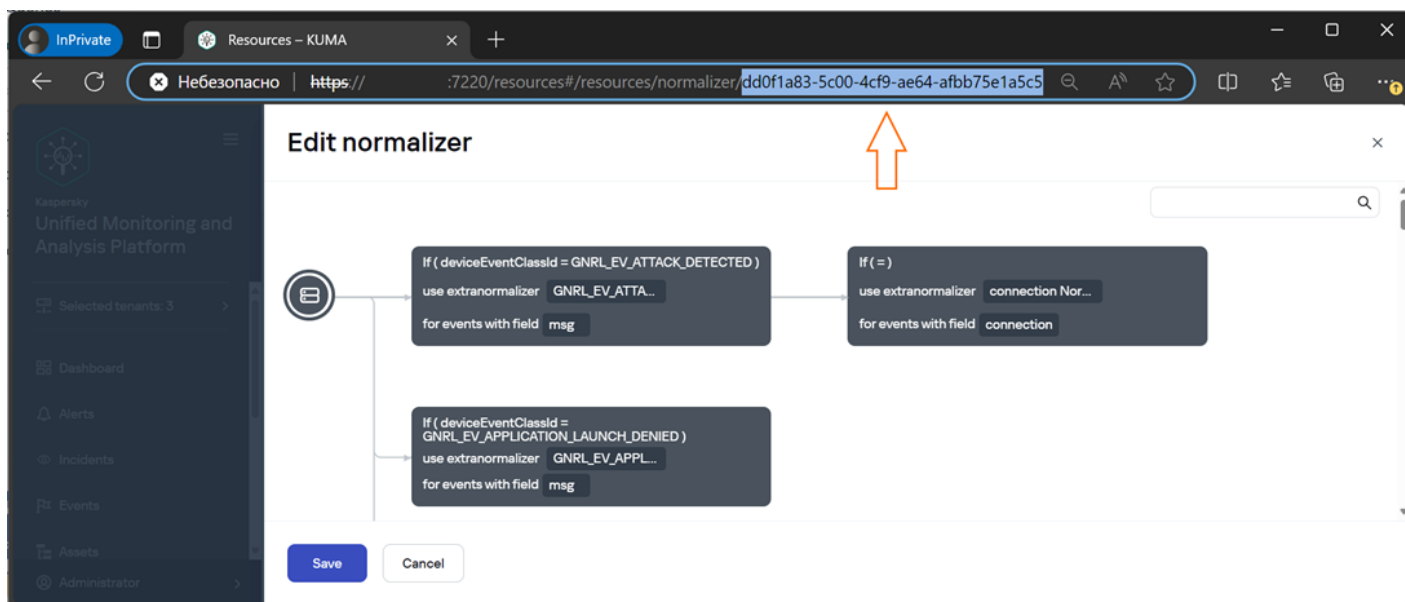


Ниже описан workaround, который позволяет получить нужное правило нормализации в виде JSON файла из MongoDB и отредактировать его, задав нужную последовательность экстранормализаторов.

Необходимые меры предосторожности:

1. Все действия по подготовке нужного правила настоятельно рекомендуется выполнять на тестовом стенде (не на продуктовой инсталляции), так как предполагается прямой доступ и запись данных в MongoDB (основную базу настроек KUMA). Нельзя исключать риск нарушения работы инсталляции KUMA из-за возможных ошибок.
2. Предварительно рекомендуется сделать выгрузку контента и бэкап самой базы: средствами kuma tools (old), по API или через утилиту mongodump.

1. Разместить на стенде KUMA правило нормализации, которое будет подлежать редактированию. Открыть его в браузере и скопировать его UUID из строки URL.



Необходимые утилиты под вашу ОС можно загрузить отсюда:

<https://www.mongodb.com/try/download/database-tools> Документация по утилитам:

<https://www.mongodb.com/docs/v4.2/reference/program/>

2. С помощью встроенной консольной утилиты mongoexport выполнить подключение к базе kuma и экспорт нужного правила нормализации в файл:

```
/opt/kaspersky/kuma/mongodb/bin/mongoexport --db=kuma --collection=resources --query='{ "_id":  
"your_normalizer_id" }' > normalizer.json
```

Пример успешного экспорта:

```
[root@kuma ~]# /opt/kaspersky/kuma/mongodb/bin/mongoexport --db=kuma --collection=resources  
--query='{ "_id": "dd0f1a83-5c00-4cf9-ae64-afbb75e1a5c5" }' > normalizer.json  
2024-06-20T11:17:45.779+0300 connected to: mongodb://localhost/  
2024-06-20T11:17:45.798+0300 exported 1 record  
[root@kuma ~]#
```

3. Открыть полученный JSON файл в редакторе, поддерживающем форматирование JSON и работу с объектами (например, Notepad++ с плагином JSTool).

— Сразу поменять uuid в полях "_id", "exportID", "id" на новый. Он должен быть уникальным в рамках всех остальных ресурсов KUMA для успешного импорта правила обратно.

Сгенерировать UUID:

```
cat /proc/sys/kernel/random/uuid
```

— Сразу поменять значение поля "name", задав новое название правила или его версию.

— Найти в структуре файла блок "extra", содержащий экстранормализаторы. Развернуть его и выполнить поиск нужного блока экстранормализации который требуется переместить.

```

normalizer.json
1 {
2   "_id": "8ce907b9-98a4-43b9-b432-b924f35eca97",
3   "exportID": "8ce907b9-98a4-43b9-b432-b924f35eca97",
4   "tenantID": "f02dbc6e-14cd-44f4-9c46-3fe9054bc210",
5   "kind": "normalizer",
6   "name": "[PRIMER] KSC from SQL Extended",
7   "description": "",
8   "createdAt": 1718625123883,
9   "updatedAt": 1718629908657,
10  "folderID": "86cfd4d8-9036-49eb-9d23-e79de6786690",
11  "userID": "56021256-4836-4356-8aee-5e1a69208095",
12  "deps": null,
13  "internal": false,
14  "repositoryPackageID": "",
15  "packageIntegrationResource": false,
16  "payload": {
17    "id": "8ce907b9-98a4-43b9-b432-b924f35eca97",
18    "name": "[PRIMER] KSC from SQL Extended",
19    "kind": "sql",
20    "expressions": [],
21    "pairDelimiter": "",
22    "kvDelimiter": "",
23    "delimiter": "",
24    "keepRaw": "always",
25    "mapping": [{
147  "enrichment": [{
261  "extra": [{
2852  "xmlKeyAttrs": [],
2853  "xmlArrayKeys": [],
2854  "samples": {
2855    "row": null,
2856    "columnar": null
2857  },
2858  "saveExtra": false,
2859  "regexGroupsCEF": false
2860  }
2861  }
2862  }

```

Annotations in the image:

- Orange arrow pointing to line 2: "Change uuid"
- Green arrow pointing to line 6: "Change name"
- Orange arrow pointing to line 17: "Change uuid"
- Green arrow pointing to line 18: "Change name"
- Orange arrow pointing to line 261: "Go to extranormalizers"

4. В блоке "extra" найти по нужным экстранормализатор, выделить и скопировать его код целиком, ориентируясь на открывающую скобку перед полем "normalizer" и соответствующую закрывающую скобку.

Extranormalizer that needs to be moved

Additional event parsing

Extra normalization conditions Normalization scheme Enrichment

Name: SQL_EV_OBJECT_DELETED

Parsing method: kv

The full code of Extranormalizer starts here

```

2548 {
2549   "normalizer": {
2550     "id": "",
2551     "name": "SQL_EV_OBJECT_DELETED", Found by name
2552     "kind": "kv",
2553     "expressions": [
2554       "\nmsg\\": \"Описание\\результата:\\s{?P<event_outcome>.*?}\\|Тип:\\s{?P<type>.*?}\\|Название:\\s{?P<object_name>.*?}\\|Пользователь:\\s{?P<user_name>.*?}\\|Объект:\\s{?P<file_path>.*?}\\|SHA256\\s{?P<sha256>.*?}\\|MD5:\\s{?P<md5>.*?}\\|.*\"],
2555     "pairDelimiter": "\n",
2556     "kvDelimiter": ":",
2557     "delimiter": "\n",
2558     "keepRaw": "never",
2559     "mapping": [{
2560       "sourceField": "Описание результата",
2561       "eventField": "EventOutcome",

```

5. Вставить скопированный код в нужное место в блоке "extra". Например, в его начале или между требуемых экстранормализаторов (зависит от нужного вам порядка их следования). Проверить, что все скобки { } на месте.

6. Сохранить получившийся JSON файл, перенести его обратно на сервер с KUMA Core и выполнить его импорт в MongoDB:

```
/opt/kaspersky/kuma/mongodb/bin/mongoimport --db kuma --collection resources --file new_normalizer.json
```

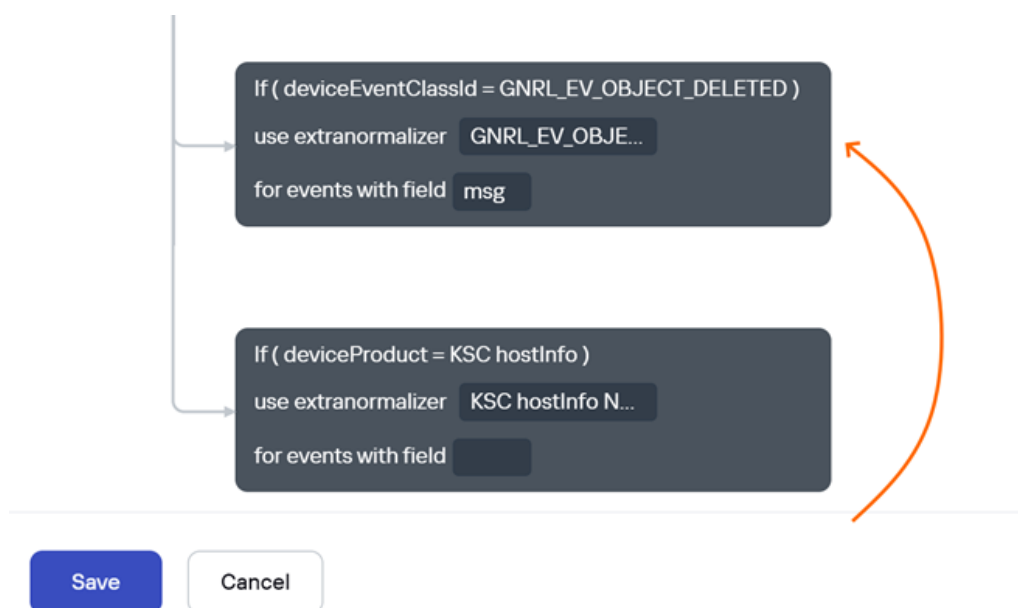
Пример успешного импорта:

```
[root@kuma ~]# /opt/kaspersky/kuma/mongodb/bin/mongoimport --db kuma --collection resources --file new_normalizer.json
2024-06-20T11:20:40.588+0300    connected to: mongodb://localhost/
2024-06-20T11:20:40.604+0300    1 document(s) imported successfully. 0 document(s) failed to import.
[root@kuma ~]#
```

7. Зайти в веб-интерфейс KUMA и проверить наличие отредактированного нормализатора (в том же тенанте и папке, т.к. они не менялись)

+ Add Duplicate Delete			
<input type="checkbox"/> Name	Kind	Updated	Created by
<input type="checkbox"/> [PRIMER] KSC from SQL Extended	sql	2024-06-20 11:23:38	Administrator

Результат: изменён порядок следования экстранормализаторов без их удаления и пересоздания вручную через веб-интерфейс



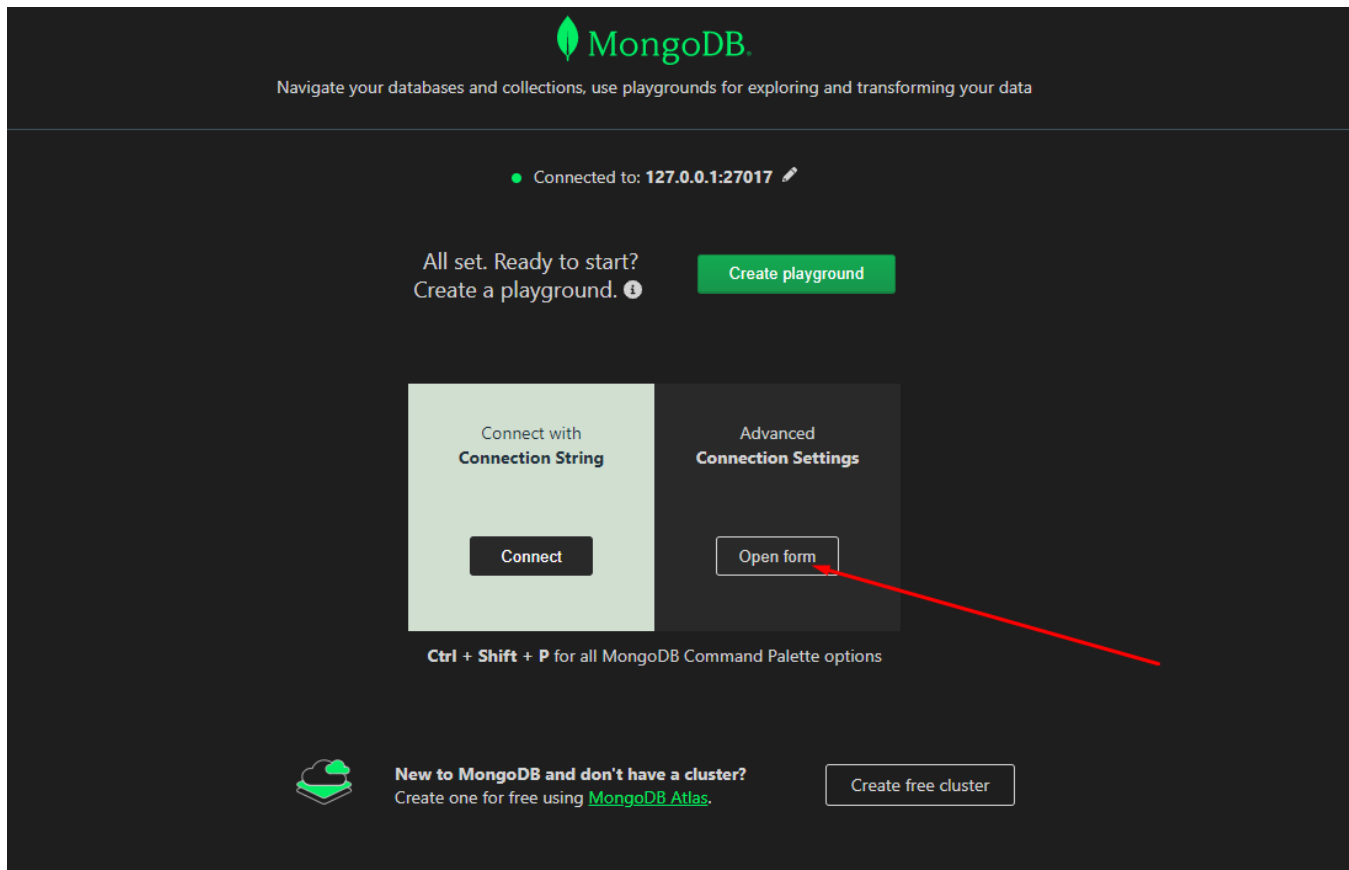
Альтернативный вариант с VS Code

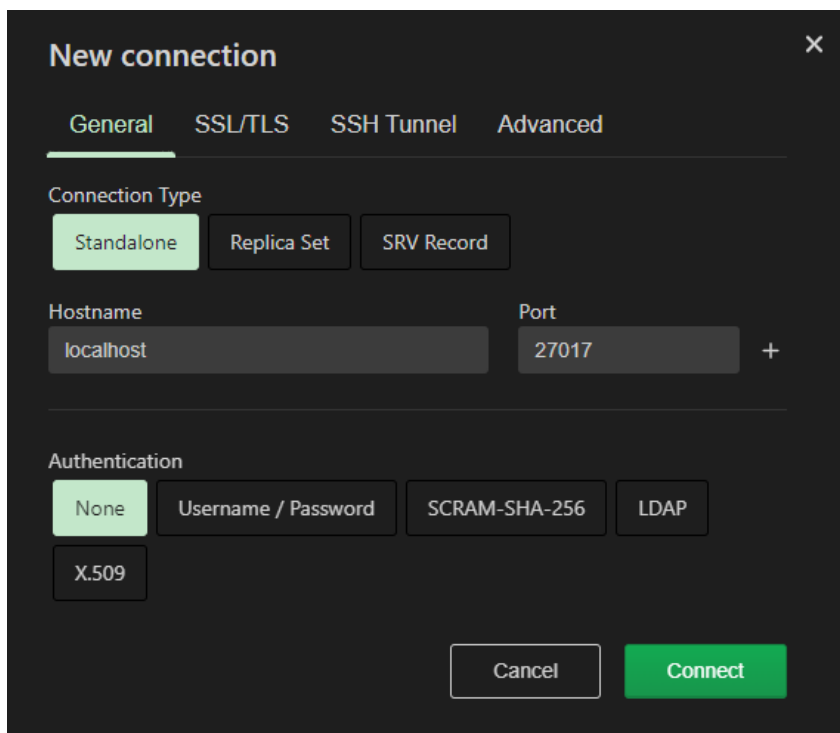
Потребуется приложение: <https://code.visualstudio.com/> и плагин для работы с MongoDB: <https://marketplace.visualstudio.com/items?itemName=mongodb.mongodb-vscode>

Также можно использовать клиент MongoDB Compass для подключения к MongoDB:
<https://www.mongodb.com/products/tools/compass>

Рекомендуется работать с копией / дубликатом ресурса, чтобы предотвратить возможные проблемы

1. Подключение к монго через SSH:





New connection [X]

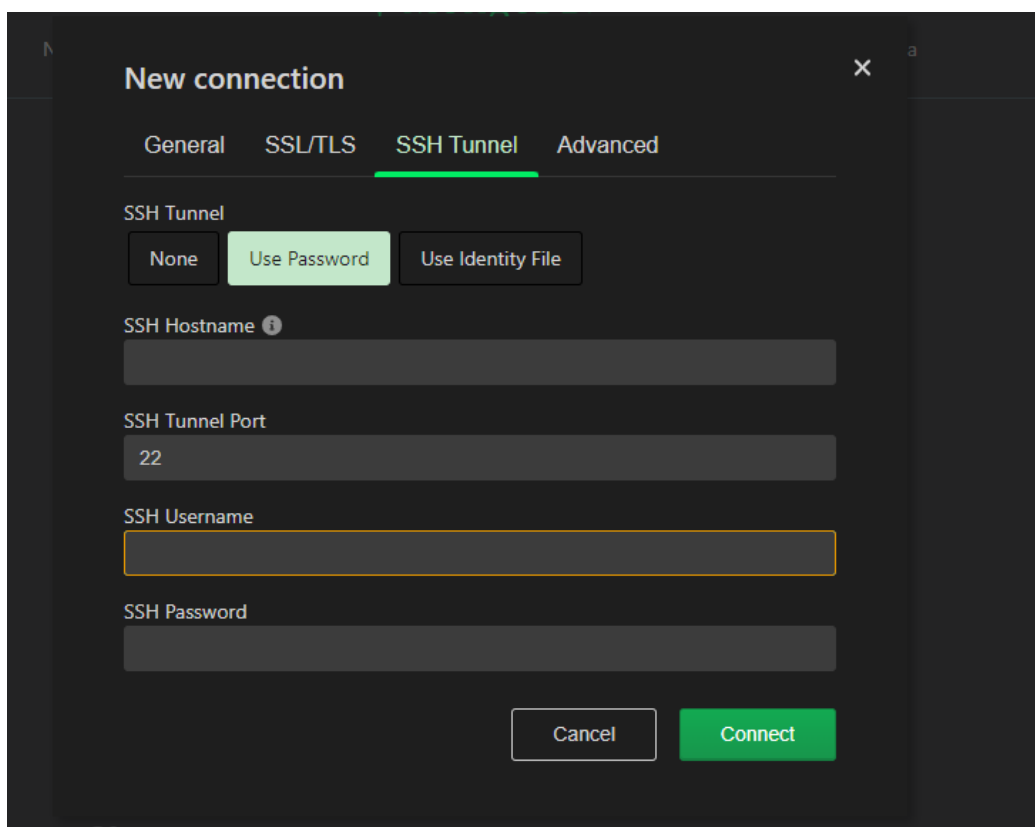
General SSL/TLS SSH Tunnel Advanced

Connection Type

Hostname Port +

Authentication

Прописываем адрес, логин и пароль для SSH:



New connection [X]

General SSL/TLS SSH Tunnel Advanced

SSH Tunnel

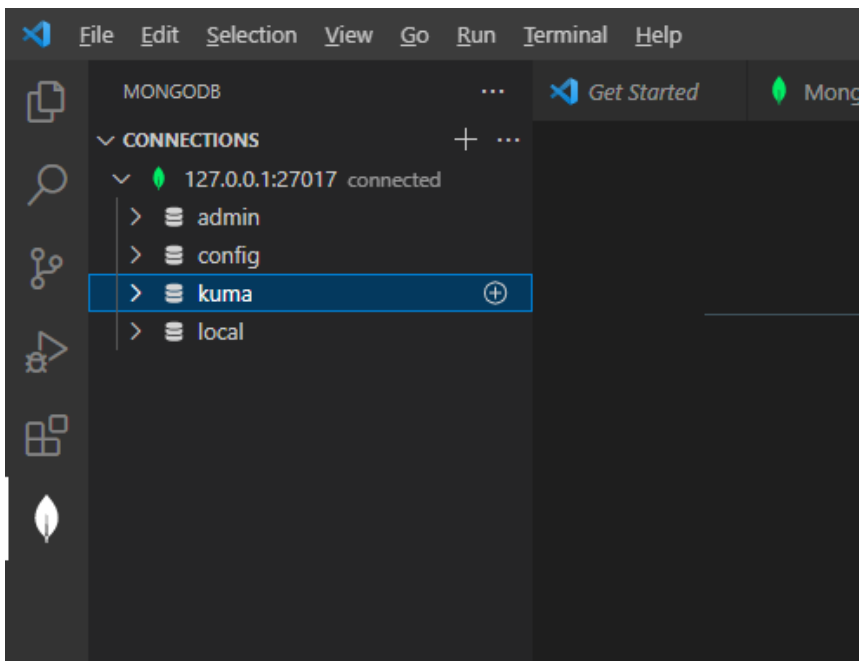
SSH Hostname ⓘ

SSH Tunnel Port

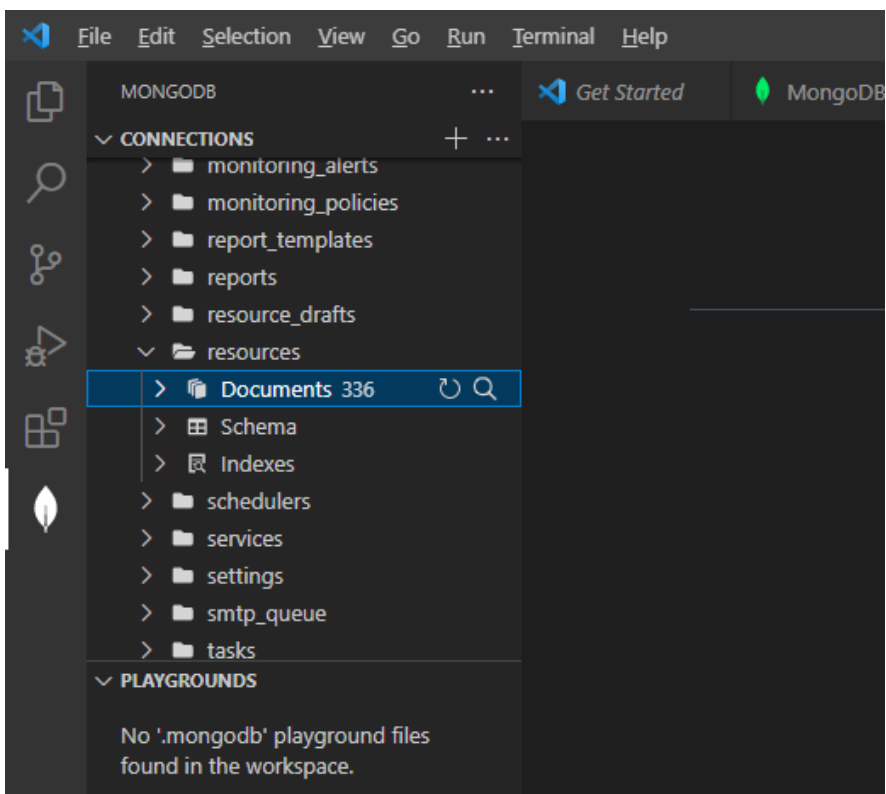
SSH Username

SSH Password

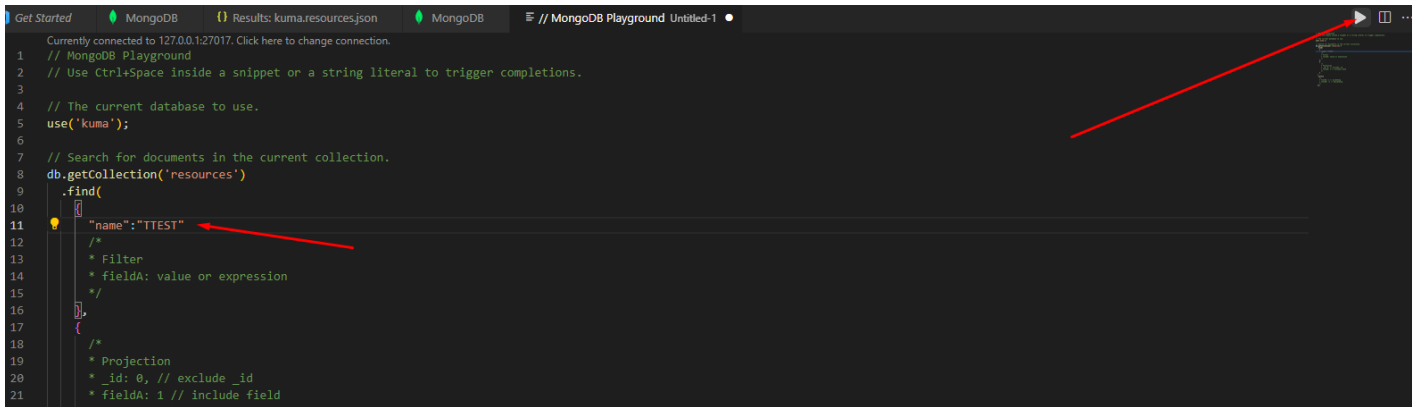
2. Заходим в БД кума



3. Переходим в коллекцию resources и нажимаем на значок поиска

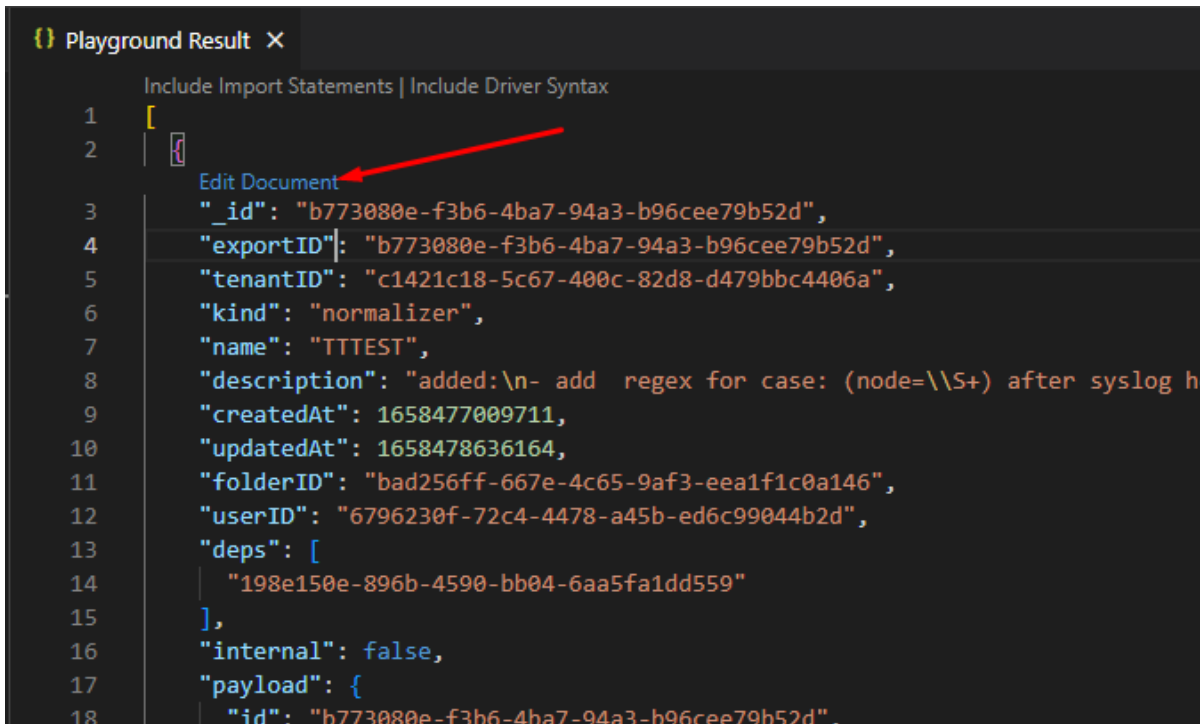


4. Ищем ресурс по имени или ID (кому как удобно):



```
1 // MongoDB Playground
2 // Use Ctrl+Space inside a snippet or a string literal to trigger completions.
3
4 // The current database to use.
5 use('kuma');
6
7 // Search for documents in the current collection.
8 db.getCollection('resources')
9   .find(
10     {
11       "name": "TTEST"
12     },
13     {
14       /*
15        * Filter
16        * fieldA: value or expression
17        */
18     },
19     {
20       /*
21        * Projection
22        * _id: 0, // exclude _id
23        * fieldA: 1 // include field
24        */
25     }
26   )
```

5. Переходим в режим редактирования и далее можно перемежать блоки JSON нормализатора и других ресурсов, как вам удобно, для сохранения используйте комбинацию клавиш Ctrl+S:



```
1 [
2   {
3     "_id": "b773080e-f3b6-4ba7-94a3-b96cee79b52d",
4     "exportID": "b773080e-f3b6-4ba7-94a3-b96cee79b52d",
5     "tenantID": "c1421c18-5c67-400c-82d8-d479bbc4406a",
6     "kind": "normalizer",
7     "name": "TTEST",
8     "description": "added:\n- add regex for case: (node=\\S+) after syslog h
9     "createdAt": 1658477009711,
10    "updatedAt": 1658478636164,
11    "folderID": "bad256ff-667e-4c65-9af3-eea1f1c0a146",
12    "userID": "6796230f-72c4-4478-a45b-ed6c99044b2d",
13    "deps": [
14      "198e150e-896b-4590-bb04-6aa5fa1dd559"
15    ],
16    "internal": false,
17    "payload": {
18      "id": "b773080e-f3b6-4ba7-94a3-b96cee79b52d",
```