

Kafka-connect-lab

Database Sink

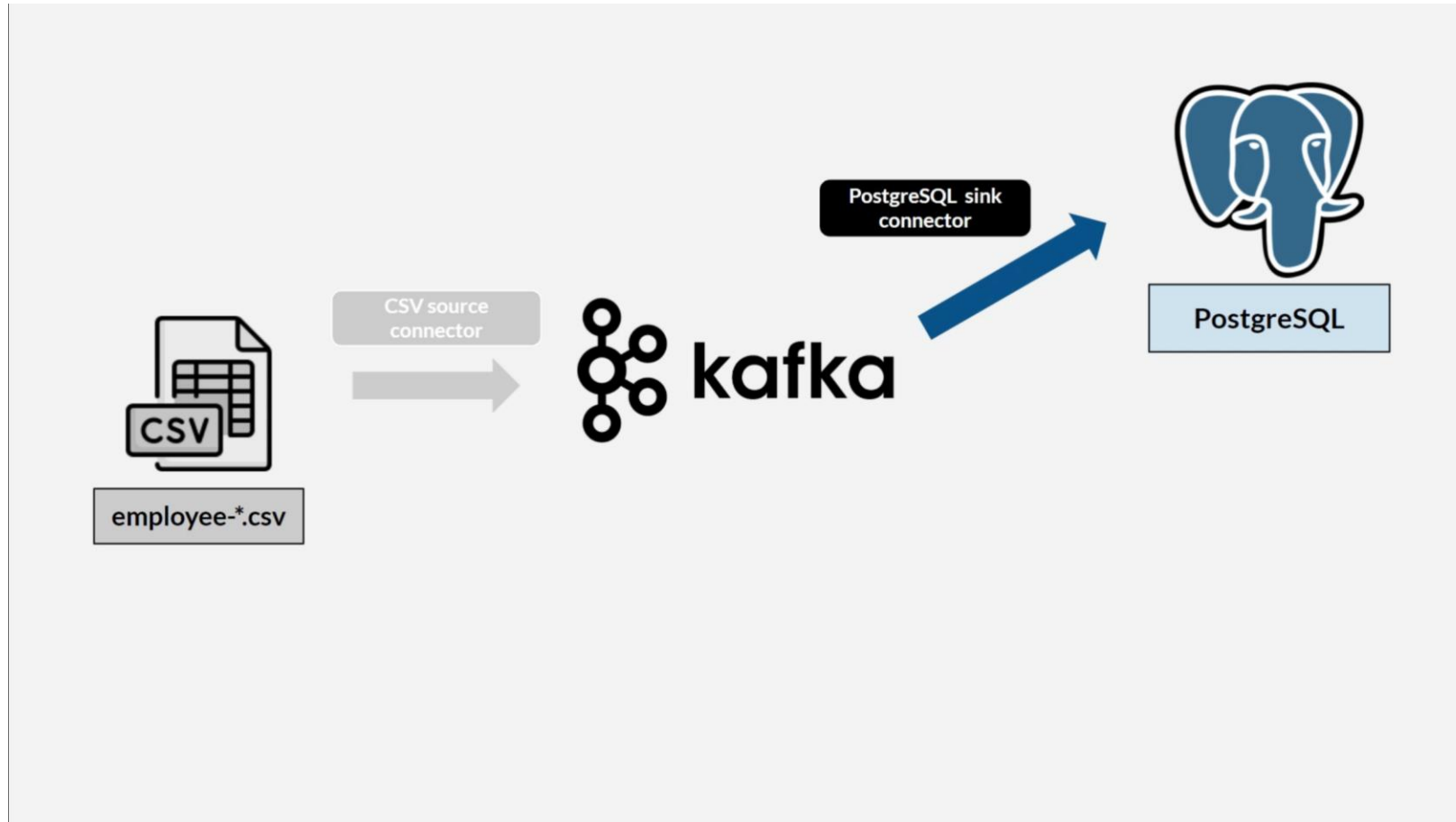
Sftp Sink

Start lab

- `docker-compose -f docker-compose-core.yml -p sample down` (remove existing containers running on same port)
- **Start Kafka connect containers:**
- `docker-compose -f docker-compose-connect.yml -p connect up -d`
- `docker-compose -f docker-compose-connect-sample.yml -p connect-sample up -d`

Database Sink

Use existing topic for csv(discussed in the session)



- Download Connector
 - <https://www.confluent.io/hub/confluentinc/kafka-connect-jdbc>
- Place into connectors folder
 - “data\kafka-connect-data\connectors”
- Restart kafka-connect container
 - `docker-compose -f docker-compose-connect.yml -p connect restart kafka-connect`

Check from postman

The screenshot shows the Postman interface with a REST client request configured to list Kafka connector plugins. The sidebar on the left shows a project tree with the following structure:

- spring kafka
 - Course - Spring Kafka 4
 - Microservices & Kafka Stream
 - Kafka Connect
 - Connectors
 - GET List connector plugins** (highlighted with a blue arrow)
 - GET List connectors (name on...
 - GET Get specific connector st...
 - GET Get specific connector c...
 - PUT Update specific connector
 - POST Restart specific connector
 - PUT Pause specific connector
 - PUT Resume specific connector
 - DEL Delete specific connector
 - Setup source connectors
 - POST Spooldir - CSV
 - POST PostgreSQL CDC - Finance
 - POST PostgreSQL CDC - Mark...
 - POST PostgreSQL - Person Ad...
 - POST HTTP - Person Address

The main panel shows the request details for the **GET** method, with the URL `{{kafkaConnectUrl}}/connector-plugins` (highlighted with a blue arrow). The **Send** button is also highlighted with a blue arrow.

The response is a JSON array of connector objects, displayed in the **Body** tab. The response status is **200 OK** with a response time of **26 ms** and a size of **1.48 KB**. The JSON response is as follows:

```
34  {
35    "type": "source",
36    "version": "0.0.0.0"
37  },
38  {
39    "class": "io.confluent.connect.jdbc.JdbcSinkConnector",
40    "type": "sink",
41    "version": "10.3.3"
42  },
43  {
44    "class": "io.confluent.connect.jdbc.JdbcSourceConnector",
45    "type": "source",
```

Blue arrows point to the `"class": "io.confluent.connect.jdbc.JdbcSinkConnector"` and `"class": "io.confluent.connect.jdbc.JdbcSourceConnector"` entries in the JSON response.

JDBC Sink Connector Configuration Properties

- https://docs.confluent.io/kafka-connect-jdbc/current/sink-connector/sink_config_options.html

Get ip from local system

```
C:\Users\bvsra>ipconfig
```



```
Windows IP Configuration
```

```
Ethernet adapter vEthernet (WSL):
```



```
Connection-specific DNS Suffix . :
```

```
Link-local IPv6 Address . . . . . : fe
```



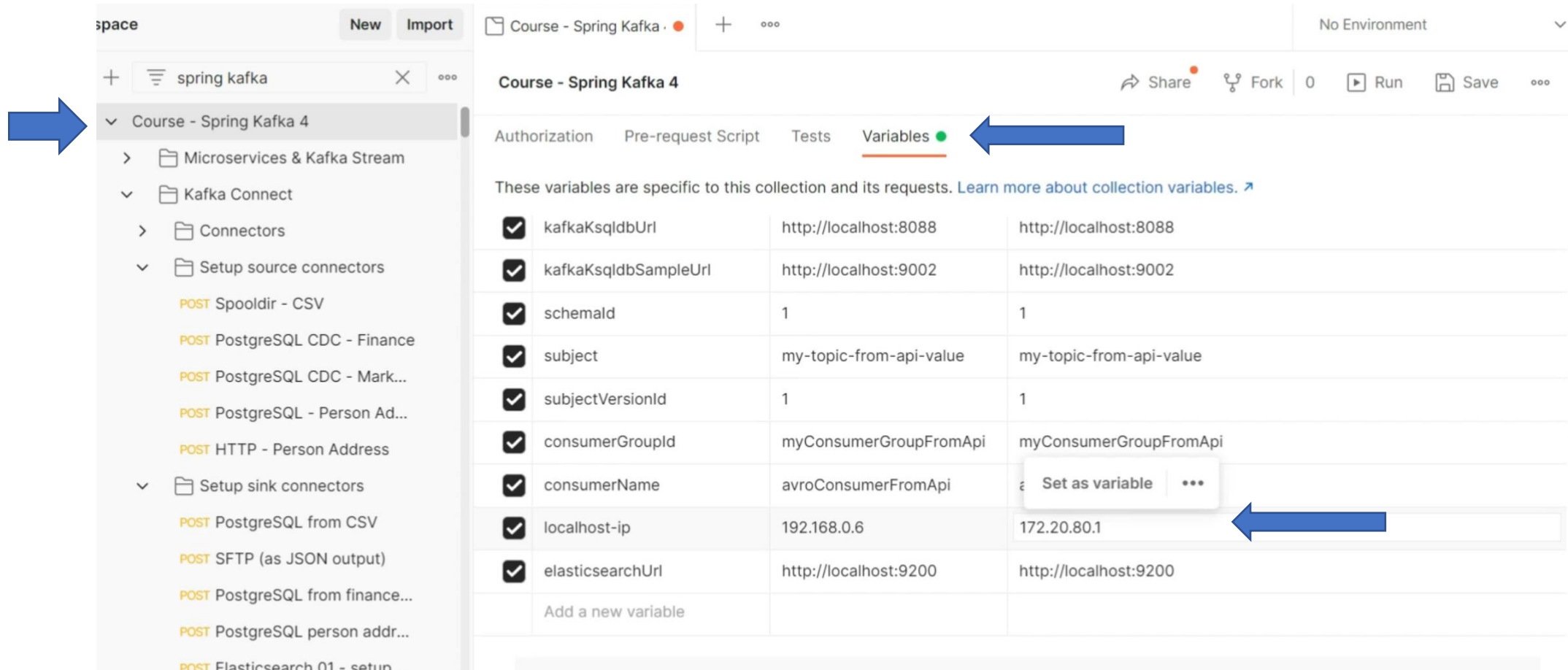
```
IPv4 Address. . . . . : 172
```



```
Subnet Mask . . . . . : 255.255.240.0
```

```
Default Gateway . . . . . :
```

Update localhost ip in postman variables



The screenshot shows the Postman interface with the 'Variables' tab selected for the 'Course - Spring Kafka 4' collection. The 'localhost-ip' variable is highlighted, and a context menu is open showing the 'Set as variable' option.

Variable	Current Value	Environment Variable
kafkaKsqlDbUrl	http://localhost:8088	http://localhost:8088
kafkaKsqlDbSampleUrl	http://localhost:9002	http://localhost:9002
schemald	1	1
subject	my-topic-from-api-value	my-topic-from-api-value
subjectVersionId	1	1
consumerGroupId	myConsumerGroupFromApi	myConsumerGroupFromApi
consumerName	avroConsumerFromApi	avroConsumerFromApi
localhost-ip	192.168.0.6	172.20.80.1
elasticsearchUrl	http://localhost:9200	http://localhost:9200

Set as variable

Create db sink connector

The screenshot displays the Postman interface for creating a PostgreSQL sink connector. The left sidebar shows the project structure under 'spring kafka', with 'Setup sink connectors' selected. The main panel shows a POST request to the endpoint `{{kafkaConnectUri}}/connectors`. The request body is a JSON object with the following configuration:

```
9 ..... "connection.password": "postgres",
10 ..... "table.name.format": "kafka_employees",
11 ..... "auto.create": true,
12 ..... "auto.evolve": true,
13 ..... "pk.mode": "record_value",
14 ..... "pk.fields": "employee_id",
15 ..... "insert.mode": "upsert"
16 ..... }
17 }
```

The response is displayed in the bottom panel, showing a JSON object with the connector name and configuration:

```
1 ..... {
2     "name": "sink-postgresql-csv",
3     "config": {
4         "connector.class": "io.confluent.connect.jdbc.JdbcSinkConnector",
5         "topics": "t-spool-dir-csv-demo",
6         "confluent.topic.bootstrap.servers": "172.20.80.1:9092",
7         "connection.url": "jdbc:postgresql://172.20.80.1:5432/postgres",
8         "connection.user": "postgres",
9         "connection.password": "postgres"
10    }
11 }
```

Check Status for specific connector

The screenshot displays a REST client interface with a sidebar on the left and a main panel on the right.

Sidebar:

- Course - Spring Kafka 4
 - Microservices & Kafka Stream
 - Kafka Connect
 - Connectors
 - GET List connector plugins
 - GET List connectors (name on...
 - GET Get specific connector st...**
 - GET Get specific connector c...
 - PUT Update specific connector
 - POST Restart specific connector
 - PUT Pause specific connector
 - PUT Resume specific connector
 - DEL Delete specific connector
 - Setup source connectors
 - POST Spoolidir - CSV
 - POST PostgreSQL CDC - Finance
 - POST PostgreSQL CDC - Mark...
 - POST PostgreSQL - Person Ad...
 - POST HTTP - Person Address

Main Panel:

Request:

- Method: GET
- URL: `{{kafkaConnectUrl}}/connectors/:connector-name/status`
- Buttons: Send, Cookies
- Tabs: Params, Authorization, Headers (7), Body, Pre-request Script, Tests, Settings

Query Params:

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Path Variables:

KEY	VALUE	DESCRIPTION	...	Bulk Edit
connector-name	sink-postgresql-csv	Description		

Response:

- Status: 200 OK, 29 ms, 321 B
- Buttons: Save Response, Copy, Search
- Tabs: Body, Cookies, Headers (4), Test Results
- Format: Pretty (selected), Raw, Preview, Visualize, JSON

Response Body (JSON):

```
1 {
2   "name": "sink-postgresql-csv",
3   "connector": {
4     "state": "RUNNING",
5     "worker_id": "kafka-connect:8083"
6   },
7   "tasks": [
8     {
```

Connect to “postgres sql” from workbench or any other specific ide

The screenshot shows the MySQL Workbench interface with a PostgreSQL connection. The Database Navigator on the left shows the schema structure, including the 'kafka_employees' table. The SQL Editor in the center contains a query to select all columns from the 'kafka_employees' table. The bottom panel displays the query results in a grid format, showing 12 rows of employee data.

Database Navigator:

- postgres - localhost:5432
 - Databases
 - postgres
 - Schemas
 - public
 - Tables
 - kafka_employees 32K

SQL Editor:

```
1=SELECT employee_id, first_name, last_name, email, gender, birth_date, salary
2 FROM public.kafka_employees;
3
```

Query Results:

employee_id	first_name	last_name	email	gender	birth_date	salary
79-8035819	Hanny	Stubs	hstubs0@github.com	F	1996-02-02	7061
24-9676514	Val	Bonnick	vbonnick1@icq.com	M	1991-09-11	2657
61-3900911	Juanita	Lanfranchi	jlanfranchi2@rediff.com	F	1992-07-08	7241
28-0495338	Elsie	Brownlea	ebrownlea3@theglobeandmail.com	F	1985-11-03	8026
13-9980006	Chris	Kettlestringes	ckettlestringes4@indiegogo.com	M	1994-12-14	1015
38-1845963	Boris	Yvens	byvens5@slideshare.net	M	1981-07-25	1798
09-2801260	Malory	Andrin	mandrin6@tripod.com	F	1999-10-23	1442
08-9859146	Kele	Orpen	korpen7@phpbb.com	M	1989-08-08	1448
91-7360438	Jennee	Oldroyde	joldroyde8@indiegogo.com	F	1981-05-28	7347
31-1588173	Marlene	Tricker	mtricker9@uol.com.br	F	1985-04-26	6351
29-6242859	Maggi	Firebrace	mfirebrace0@sphinn.com	F	1999-05-02	8200
98-3021022	Nichol	Deverall	ndeverall1@voutube.com	F	1982-05-25	8584

These entries already exist in my db. You will have no records by default

- Add sample csv file into inputs directory (find files inside kafka-connect-samples folder)
 - *data\kafka-connect-data\inputs*
 - employee-sample-3.csv
 - employee-sample-4.csv
- After 10 seconds check in db

Run the query and check in db

(you can now find new records in the db. Check by adding more sample csv files)

The screenshot shows a database management tool interface. The top menu bar includes File, Edit, Navigate, Search, SQL Editor, Database, Window, and Help. The main window is divided into several panes:

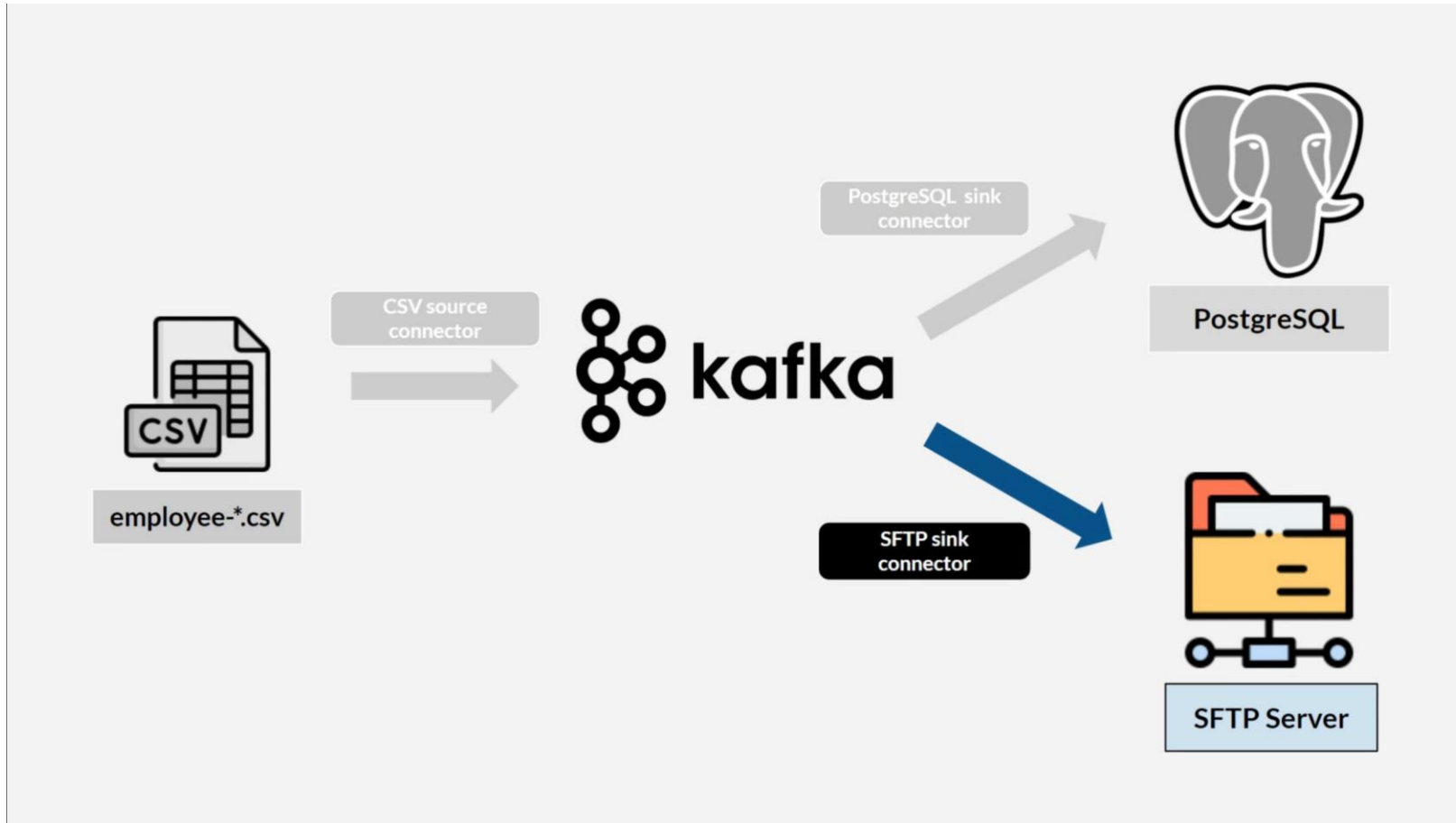
- Database Navigator:** Shows the database structure for 'postgres - localhost:5432'. It includes a tree view with 'Databases', 'Schemas', 'public', 'Tables', 'Views', 'Materialized Views', 'Indexes', 'Functions', 'Sequences', 'Data types', 'Aggregate functions', 'Event Triggers', 'Extensions', 'Storage', 'System Info', 'Roles', 'Administer', and 'System Info'. The 'kafka_employees' table is selected under 'public' > 'Tables'.
- SQL Editor:** Contains the following SQL query:

```
1=SELECT employee_id, first_name, last_name, email, gender, birth_date, salary
2 FROM public.kafka_employees;
3
```
- Results Grid:** Displays the results of the query in a table format. The columns are: employee_id, first_name, last_name, email, gender, birth_date, and salary. The results show 20 rows of employee data.
- Project - General:** Shows the project settings, including 'Name' and 'DataSource'.

The bottom status bar indicates '200' rows, '20' rows fetched, and a fetch time of '7ms (1ms fetch)' on 'Apr 05, 16:2'.

employee_id	first_name	last_name	email	gender	birth_date	salary
79-8035819	Hanny	Stubs	hstubs0@github.com	F	1996-02-02	7061
24-9676514	Val	Bonnick	vbonnick1@icq.com	M	1991-09-11	2657
61-3900911	Juanita	Lanfranci	jlanfranci2@rediff.com	F	1992-07-08	7241
28-0495338	Elsie	Brownlea	ebrownlea3@theglobeandmail.com	F	1985-11-03	8026
13-9980006	Chris	Kettlestringes	ckettlestringes4@indiegogo.com	M	1994-12-14	1015
38-1845963	Boris	Yvens	byvens5@slideshare.net	M	1981-07-25	1798
09-2801260	Malory	Andrin	mandrin6@tripod.com	F	1999-10-23	1442
08-9859146	Kele	Orpen	korpen7@phpbb.com	M	1989-08-08	1448
91-7360438	Jennee	Oldroyde	joldroyde8@indiegogo.com	F	1981-05-28	7347
31-1588173	Marlene	Tricker	mtricker9@uol.com.br	F	1985-04-26	6351
29-6242859	Maggi	Firebrace	mfirebrace0@sphinn.com	F	1999-05-02	8200
98-3021022	Nichol	Deverall	ndeverall1@voutube.com	F	1982-05-25	8584

SFTP Sink



- Download SFTP Sink Connector
 - <https://www.confluent.io/hub/confluentinc/kafka-connect-sftp>
- Place into connectors folder
 - “data\kafka-connect-data\connectors”
- Restart kafka-connect container
 - `docker-compose -f docker-compose-connect.yml -p connect restart kafka-connect`

Verify connectors from postman

The image shows the Postman application interface. On the left is a sidebar with a collection named 'spring kafka'. Under the 'Connectors' folder, the request 'GET List connector plugins' is selected. The main panel displays the details of this request. The URL is `{{kafkaConnectUrl}}/connector-plugins`. The 'Params' tab is active, showing a table for query parameters.

KEY	VALUE	DESCRIPTION
Key	Value	Description

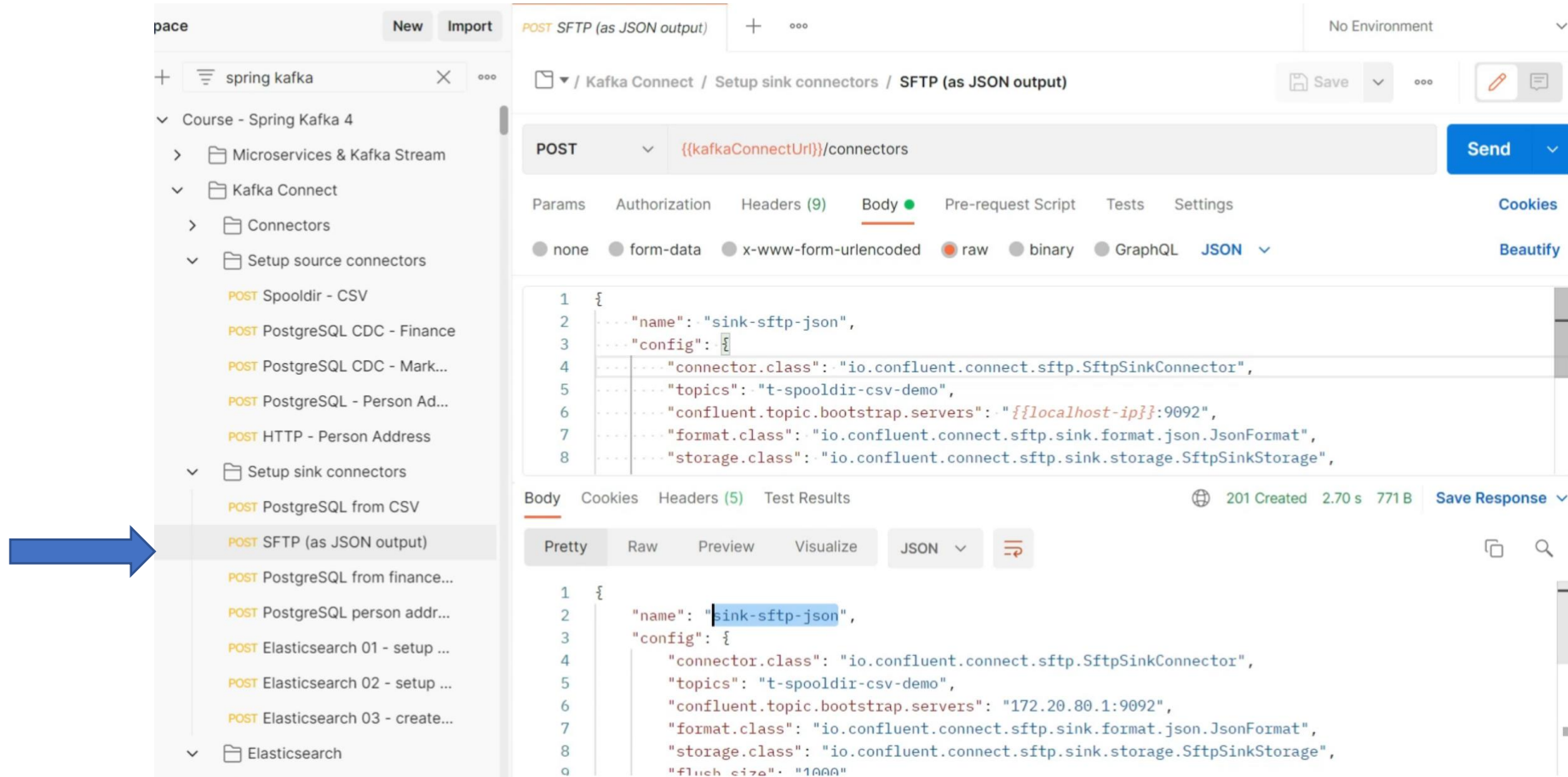
Below the table, the 'Body' tab is selected, showing a JSON response in 'Pretty' format. The response status is 200 OK, with a response time of 22 ms and a size of 2.17 KB.

```
51  },
52  {
53    "class": "io.confluent.connect.sftp.SftpCsvSourceConnector",
54    "type": "source",
55    "version": "0.0.0.0"
56  },
57  {
58    "class": "io.confluent.connect.sftp.SftpGenericSourceConnector",
59    ...
```


SFTP Sink Connector Configuration Properties

- https://docs.confluent.io/kafka-connect-sftp/current/sink-connector/configuration_options.html

Create sftp sink connector



The screenshot displays the Postman interface for creating an SFTP sink connector. A blue arrow points to the 'POST SFTP (as JSON output)' item in the left sidebar. The main panel shows the request details for the endpoint `{{kafkaConnectUrl}}/connectors`.

Request Details:

- Method:** POST
- URL:** `{{kafkaConnectUrl}}/connectors`
- Body Type:** JSON

Request Body (JSON):

```
1 {
2   "name": "sink-sftp-json",
3   "config": {
4     "connector.class": "io.confluent.connect.sftp.SftpSinkConnector",
5     "topics": "t-spool-dir-csv-demo",
6     "confluent.topic.bootstrap.servers": "172.20.80.1:9092",
7     "format.class": "io.confluent.connect.sftp.sink.format.json.JsonFormat",
8     "storage.class": "io.confluent.connect.sftp.sink.storage.SftpSinkStorage",
9   }
10 }
```

Response Details:

- Status:** 201 Created
- Time:** 2.70 s
- Size:** 771 B

Response Body (JSON):

```
1 {
2   "name": "sink-sftp-json",
3   "config": {
4     "connector.class": "io.confluent.connect.sftp.SftpSinkConnector",
5     "topics": "t-spool-dir-csv-demo",
6     "confluent.topic.bootstrap.servers": "172.20.80.1:9092",
7     "format.class": "io.confluent.connect.sftp.sink.format.json.JsonFormat",
8     "storage.class": "io.confluent.connect.sftp.sink.storage.SftpSinkStorage",
9     "flush.size": "1000"
10  }
```

Check Status

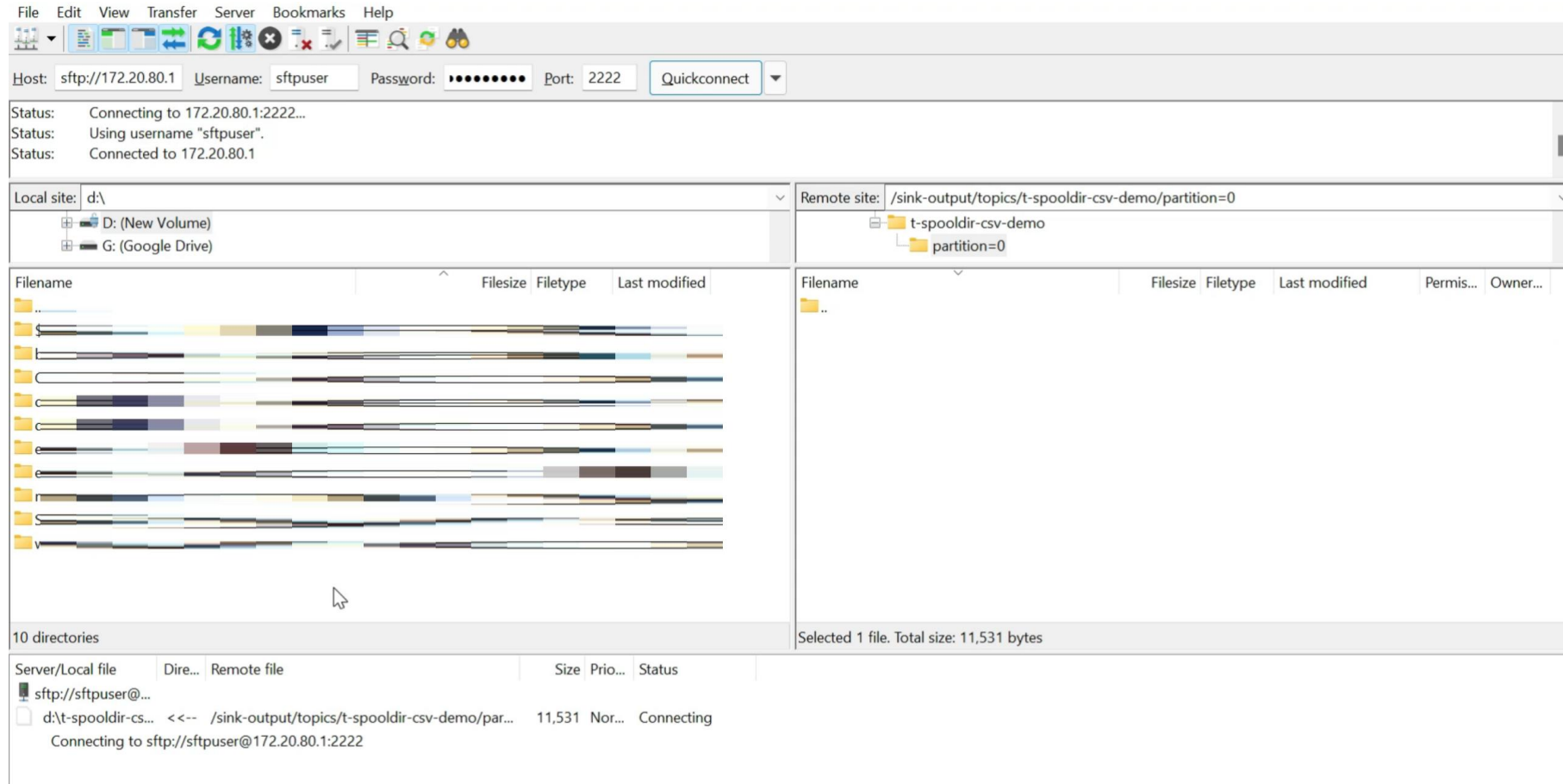
The screenshot shows the Postman interface with a REST client set up to check the status of a Kafka connector. The left sidebar shows a collection named 'spring kafka' with a folder 'Connectors' containing several GET requests. The selected request is 'GET Get specific connector st...'. The main panel shows the request details:

- Method:** GET
- URL:** `{{kafkaConnectUrl}}/connectors/:connector-name/status`
- Path Variables:** A table with columns KEY, VALUE, and DESCRIPTION. The row for 'connector-name' has the value 'sink-sftp-json', which is highlighted with a blue arrow.
- Response:** The status is 200 OK, 26 ms, 316 B. The response body is shown in JSON format:

```
1 {  
2   "name": "sink-sftp-json",  
3   "connector": {  
4     "state": "RUNNING",  
5     "worker_id": "kafka-connect:8083"  
6   },  
7   "tasks": [  
8     {
```

Use any sftp connector

(username and password find from the properties set in postman while creating sftp connector)



- Add sample csv file into inputs directory (find files inside kafka-connect-samples folder)
 - *data\kafka-connect-data\inputs*
 - employee-sample-7.csv
 - employee-sample-8.csv
- After 10 seconds check in db

Verify data generated

The screenshot shows the WinSCP interface with the following details:

- Host:** sftp://172.20.80.1
- Username:** sftpuser
- Password:** [masked]
- Port:** 2222
- Status:** Connecting to 172.20.80.1:2222...
Using username "sftpuser".
Connected to 172.20.80.1
- Local site:** d:\
- Remote site:** /sink-output/topics/t-spool-dir-csv-demo/partition=0
- Local site file list:**

Filename	Filesize	Filetype	Last modified
..			
\$R			
ba			
Co			
de			
dc			
eb			
ec			
my			
Sy			
wc			
- Remote site file list:**

Filename	Filesize	Filetype	Last modified	Permis...	Owner...
..					
t-spool-dir-csv-demo+0+0000000000.json	15,563	.JSON File	2022-04-05 17:04:16	-rw-r--...	1000 1...
- Selected file:** t-spool-dir-csv-demo+0+0000000000.json
- Selected file size:** 15,563 bytes
- Directories:** 10 directories
- Server/Local file:** sftp://sftpuser@...
d:\t-spool-dir-cs... <<- /sink-output/topics/t-spool-dir-csv-demo/par...
Connecting to sftp://sftpuser@172.20.80.1:2222

A red arrow points to the selected JSON file in the remote site list.

Download and check this json file, csv files are converted to json and stored inside sftp server

References

- <https://docs.confluent.io/platform/current/connect/userguide.html>
- https://developer.confluent.io/learn-kafka/kafka-connect/intro/?_ga=2.55576299.1562784863.1655844784-892683210.1655397866&_gac=1.259964536.1655848660.Cj0KCQjw2MWVBhCQARIsAljbwoMz0SyR61XgsR9Y7jnJ32-QDMLsVhp7wEgcApYklKjcpFo2Lh0RmilaAt-IEALw_wcB