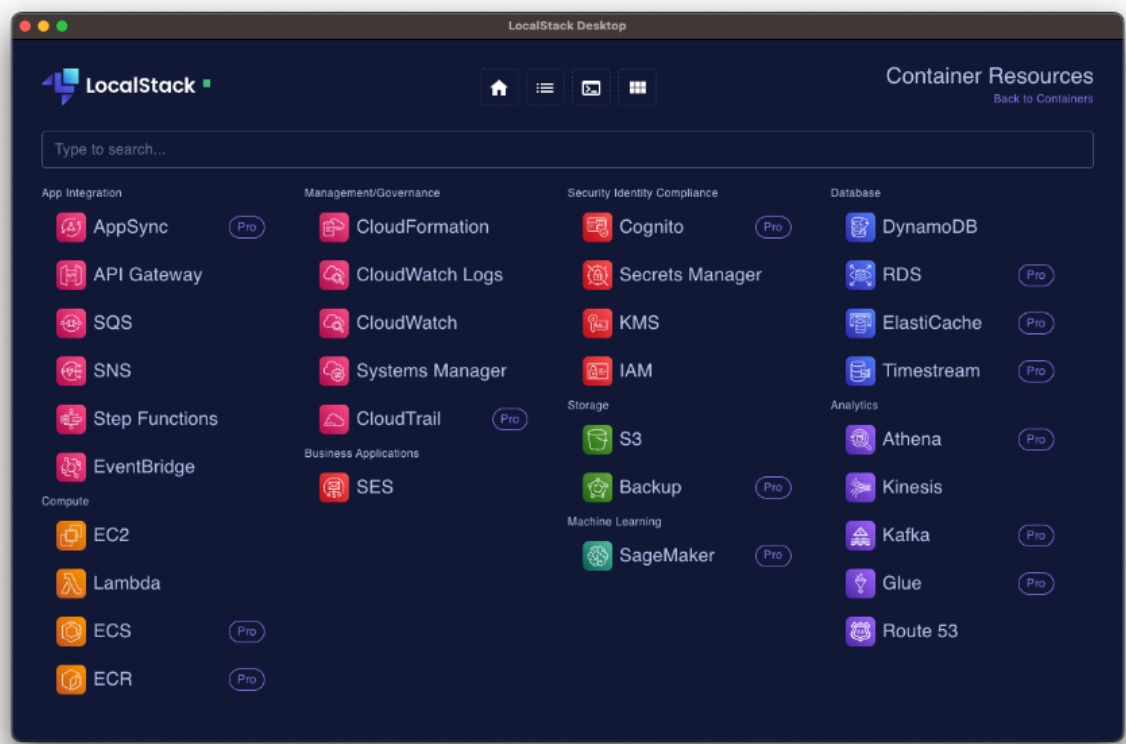


- <https://www.localstack.cloud/>
- <https://docs.localstack.cloud/user-guide/tools/localstack-desktop/>

- Kafka / ELasticsearch(ElasticCache)
- Glue/Athena/SageMaker AI
 - AWS AI ~ ()
 - - - OFF

Resource browser

You can also create, modify, delete and read all of your resources from the Resource Browser tab, having the same experience that you would have using it in our [web application](#)



- OpenStack AWS .

AWS S3 Stream S3 .

LocalStack

```

version: '3'
services:
  localstack:
    image: localstack/localstack:3.0.2
    ports:
      - "4567:4566"
    environment:
      - SERVICES=s3:4566
      - HOSTNAME=localstack
      - HOSTNAME_EXTERNAL=localstack
      - DEFAULT_REGION=us-east-1
  aws-cli:
    image: amazon/aws-cli
    depends_on:
      - localstack
    volumes:
      - ./testbucket:/aws
      - /opt/localstack:/tmp/localstack-s3-storage
      - "/var/run/docker.sock:/var/run/docker.sock"
    environment:
      - AWS_ACCESS_KEY_ID=test
      - AWS_SECRET_ACCESS_KEY=test
      - AWS_DEFAULT_REGION=us-east-2
    entrypoint: /bin/sh -c
    command: >
      "
      touch testData.csv
      touch testDataOne.csv
      touch testDataTwo.csv
      touch testDataThree.csv
      aws --endpoint-url=http://localstack:4566 s3api create-bucket --bucket my-bucket --region us-east-1
      aws --endpoint-url=http://localstack:4566 s3 cp testData.csv s3://my-bucket//testData.csv
      aws --endpoint-url=http://localstack:4566 s3 cp testDataOne.csv s3://my-bucket//testDataTwo.csv
      aws --endpoint-url=http://localstack:4566 s3 cp testDataTwo.csv s3://my-bucket//testDataTwo.csv
      aws --endpoint-url=http://localstack:4566 s3 cp testDataThree.csv s3://my-bucket//testDataThree.csv
      "

```

- docker-compose up -d

```

2024-01-10 11:24:19
2024-01-10 11:24:19 LocalStack version: 3.0.2
2024-01-10 11:24:19 LocalStack build date: 2023-11-29
2024-01-10 11:24:19 LocalStack build git hash: 60518e11
2024-01-10 11:24:19
2024-01-10 11:24:21 2024-01-10T02:24:21.008 WARN --- [ MainThread] localstack.deprecations :
DEFAULT_REGION is deprecated (since 0.12.7) and will be removed in upcoming releases of LocalStack! LocalStack
now has full multi-region support. This option has no effect. Please remove it from your configuration.
2024-01-10 11:24:21 2024-01-10T02:24:21.008 WARN --- [ MainThread] localstack.deprecations :
HOSTNAME_EXTERNAL is deprecated (since 2.0.0) and will be removed in upcoming releases of LocalStack! This
configuration will be migrated to LOCALSTACK_HOST
2024-01-10 11:24:21 2024-01-10T02:24:21.898 INFO --- [-functhread4] hypercorn.error : Running on
https://0.0.0.0:4566 (CTRL + C to quit)
2024-01-10 11:24:21 2024-01-10T02:24:21.898 INFO --- [-functhread4] hypercorn.error : Running on
https://0.0.0.0:4566 (CTRL + C to quit)
2024-01-10 11:24:21 2024-01-10T02:24:21.942 INFO --- [ MainThread] localstack.utils.bootstrap : Execution of
"start_runtime_components" took 908.67ms
2024-01-10 11:24:21 Ready.
2024-01-10 11:24:32 2024-01-10T02:24:32.335 INFO --- [ asgi_gw_0] localstack.utils.bootstrap : Execution of
"_load_service_plugin" took 1108.37ms
2024-01-10 11:24:32 2024-01-10T02:24:32.336 INFO --- [ asgi_gw_0] localstack.utils.bootstrap : Execution of
"require" took 1109.21ms
2024-01-10 11:24:32 2024-01-10T02:24:32.889 INFO --- [ asgi_gw_0] localstack.request.aws : AWS s3.
CreateBucket => 200
2024-01-10 11:24:36 2024-01-10T02:24:36.327 INFO --- [ asgi_gw_1] localstack.request.aws : AWS s3.
PutObject => 200
2024-01-10 11:24:38 2024-01-10T02:24:38.670 INFO --- [ asgi_gw_0] localstack.request.aws : AWS s3.
PutObject => 200
2024-01-10 11:24:39 2024-01-10T02:24:39.754 INFO --- [ asgi_gw_1] localstack.request.aws : AWS s3.
PutObject => 200
2024-01-10 11:24:40 2024-01-10T02:24:40.944 INFO --- [ asgi_gw_0] localstack.request.aws : AWS s3.
PutObject => 200

```

AWS CLI S3

LocalStack AWS .

S3 S3

AWS CLI s3api

- : `aws --endpoint-url=http://localhost:4567 s3api create-bucket --bucket my-bucket --region us-east-1`
- : `aws --endpoint-url=http://localhost:4567 s3api list-buckets`
- : `aws --endpoint-url=http://localhost:4567 s3 ls s3://my-bucket/`

more api : <https://docs.aws.amazon.com/cli/latest/reference/s3api/>

LocalStack

```

root@DESKTOP-K1IEP1L:/mnt/c/Code/Webnori/java-labs/infra# aws --endpoint-url=http://localhost:4567 s3 ls
s3://my-bucket//
2024-01-10 11:53:32          0 testData.csv
2024-01-10 11:53:34          0 testDataThree.csv
2024-01-10 11:53:34          0 testDataTwo.csv

```

MockServer (INFO) .

Docker Desktop

Update to latest

Q. Search for local and remote images, containers, and more...

Ctrl+K

Containers

Images

Volumes

Dev Environments BETA

Docker Scout EARLY ACCESS


Learning Center


Extensions

Disk usage

Add Extensions

infra-localstack-1 localstack/localstack:3.0.2

7d19c6467733 

4567:4566 

Logs

Inspect

Terminal

Files

Stats

2024-01-10 15:33:20

2024-01-10 15:33:20 LocalStack version: 3.0.2

2024-01-10 15:33:20 LocalStack build date: 2023-11-29

2024-01-10 15:33:20 LocalStack build git hash: 60518e11

2024-01-10 15:33:20

2024-01-10 15:33:20 2024-01-10T06:33:20.931 WARN --- [MainThread] localstack.deprecations : DEFAULT_REGION is deprecated (since 0.12.7) and will be re

upport. This option has no effect. Please remove it from your configuration.

2024-01-10 15:33:20 2024-01-10T06:33:20.931 WARN --- [MainThread] localstack.deprecations : HOSTNAME_EXTERNAL is deprecated (since 2.0.0) and will be

to LOCALSTACK_HOST

2024-01-10 15:33:21 2024-01-10T06:33:21.548 INFO --- [-functhread4] hypercorn.error : Running on https://0.0.0.0:4566 (CTRL + C to quit)

2024-01-10 15:33:21 2024-01-10T06:33:21.548 INFO --- [-functhread4] hypercorn.error : Running on https://0.0.0.0:4566 (CTRL + C to quit)

2024-01-10 15:33:21 2024-01-10T06:33:21.848 INFO --- [MainThread] localstack.utils.bootstrap : Execution of "start_runtime_components" took 905.15ms

2024-01-10 15:33:21 Ready.

2024-01-10 15:33:23 2024-01-10T06:33:23.997 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.CreateBucket => 200

2024-01-10 15:33:24 2024-01-10T06:33:24.588 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PutObject => 200

2024-01-10 15:33:25 2024-01-10T06:33:25.190 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PutObject => 200

2024-01-10 15:33:25 2024-01-10T06:33:25.789 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PutObject => 200

2024-01-10 15:33:26 2024-01-10T06:33:26.433 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PutObject => 200

2024-01-10 15:36:04 2024-01-10T06:36:04.161 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PostObject => 404 (NoSuchBucket)

2024-01-10 15:36:06 2024-01-10T06:36:06.170 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.ListObjects => 404 (NoSuchBucket)

2024-01-10 15:38:57 2024-01-10T06:38:57.232 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PostObject => 404 (NoSuchBucket)

2024-01-10 15:38:59 2024-01-10T06:38:59.116 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.ListObjects => 404 (NoSuchBucket)

2024-01-10 15:47:10 2024-01-10T06:47:10.221 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PostObject => 404 (NoSuchBucket)

2024-01-10 15:47:12 2024-01-10T06:47:12.354 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.ListObjects => 404 (NoSuchBucket)

2024-01-10 15:49:06 2024-01-10T06:49:06.435 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.CreateBucket => 200

2024-01-10 15:51:41 2024-01-10T06:51:41.078 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PostObject => 404 (NoSuchBucket)

2024-01-10 15:51:43 2024-01-10T06:51:43.161 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.ListObjects => 404 (NoSuchBucket)

2024-01-10 15:52:42 2024-01-10T06:52:42.779 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.CreateBucket => 200

2024-01-10 15:52:56 2024-01-10T06:52:56.906 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PostObject => 404 (NoSuchBucket)

2024-01-10 15:52:59 2024-01-10T06:52:59.149 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.ListObjects => 404 (NoSuchBucket)

2024-01-10 15:57:46 2024-01-10T06:57:46.085 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PostObject => 404 (NoSuchBucket)

2024-01-10 15:57:48 2024-01-10T06:57:48.192 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.ListObjects => 404 (NoSuchBucket)

2024-01-10 16:00:17 2024-01-10T07:00:17.749 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PostObject => 404 (NoSuchBucket)

2024-01-10 16:00:19 2024-01-10T07:00:19.748 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.ListObjects => 404 (NoSuchBucket)

2024-01-10 16:24:41 2024-01-10T07:24:41.213 INFO --- [asgi_gw_0] localstack.request.aws : AWS s3.PostObject => 404 (NoSuchBucket)

S3

```
uploadSource
    .throttle(processCountPerSec, Duration.ofSeconds(1))
    .runForeach(pair -> {
        ByteString byteString = pair.first();
        long index = pair.second();
        String dynamicFileKey = fileKey + index; // fileKey

        Source.single(byteString)
            .runWith(S3.multipartUpload(bucketName, dynamicFileKey)
                .withAttributes(S3Attributes.settings(s3Settings)), materializer)
            .thenAccept(result -> {
                LocalDateTime now = LocalDateTime.now();
                DateTimeFormatter formatter = DateTimeFormatter.ofPattern("HH:mm:ss");
                String formattedNow = now.format(formatter);
                System.out.println(formattedNow + " Upload complete: " + result.location());
                greetActor.tell("hello", null);
            })
            .exceptionally(throwable -> {
                System.err.println("Upload failed: " + throwable.getMessage());
                return null;
            });
    }, materializer);
```

```

Try Upload
Wait for UploadCompleted
23:33:50 Upload complete: http://my-bucket.s3.localhost.localstack.cloud:4566/example.txt11
23:33:50 Upload complete: http://my-bucket.s3.localhost.localstack.cloud:4566/example.txt10
23:33:50 Upload complete: http://my-bucket.s3.localhost.localstack.cloud:4566/example.txt12
23:33:51 Upload complete: http://my-bucket.s3.localhost.localstack.cloud:4566/example.txt13
23:33:52 Upload complete: http://my-bucket.s3.localhost.localstack.cloud:4566/example.txt14
23:33:53 Upload complete: http://my-bucket.s3.localhost.localstack.cloud:4566/example.txt15
23:33:54 Upload complete: http://my-bucket.s3.localhost.localstack.cloud:4566/example.txt16
23:33:55 Upload complete: http://my-bucket.s3.localhost.localstack.cloud:4566/example.txt17
23:33:56 Upload complete: http://my-bucket.s3.localhost.localstack.cloud:4566/example.txt18
23:33:57 Upload complete: http://my-bucket.s3.localhost.localstack.cloud:4566/example.txt19
Try Download
23:33:57 Downloaded: example.txt19
23:33:57 Downloaded: example.txt10
23:33:57 Downloaded: example.txt17
23:33:57 Downloaded: example.txt11
23:33:57 Downloaded: example.txt12
23:33:57 Downloaded: example.txt14
23:33:57 Downloaded: example.txt16
23:33:57 Downloaded: example.txt18
23:33:57 Downloaded: example.txt15
23:33:57 Downloaded: example.txt13

```

- AkkaStream TPS ~ .

ReactiveStream S3

TPS .

Stack(Kafka,S3,Nosql....) ReactiveStream .

Stream .

- stream : Flow backpressure . ()
 - reactive stream case :
 - s3 kafka nosql
 - kafka s3 :

The screenshot shows an IDE with a project structure on the left, code in the center, and test results at the bottom.

Project Structure:

- alpikka
 - kafka
 - KafkaTest
 - s3
 - S3IOTest
 - webflux
 - actor
 - model
 - SubScribeActor
 - BasicGuideTest
 - README.md
 - SpringwebApplicationTests
 - resources
 - application.properties
 - factorial.conf
 - hashgroup.conf
 - logback.xml
 - master-test.conf

Code Snippet:

```

.throttle(processCountPerSec, Duration.ofSeconds(1)) .source(ByteString.captureOf(2))
.runWith(S3.multipartUpload(bucketName, fileKey)
    .withAttributes(S3Attributes.settings(s3Settings)), materializer) CompletionStage<MultipartUploadResult>
.thenAccept(result -> {
    System.out.println("Upload complete: " + result.location());
    greetActor.tell(msg("hello", sender.null));
}) CompletionStage<Void>
.exceptionally(throwable -> {
    System.err.println("Upload failed: " + throwable.getMessage());
    return null;
});

System.out.println("Wait for UploadCompleted");

// 액터 수신검증을 위한 원본검증
probe.expectMsg(Duration.ofSeconds(15), obj("world"));

```

Test Results:

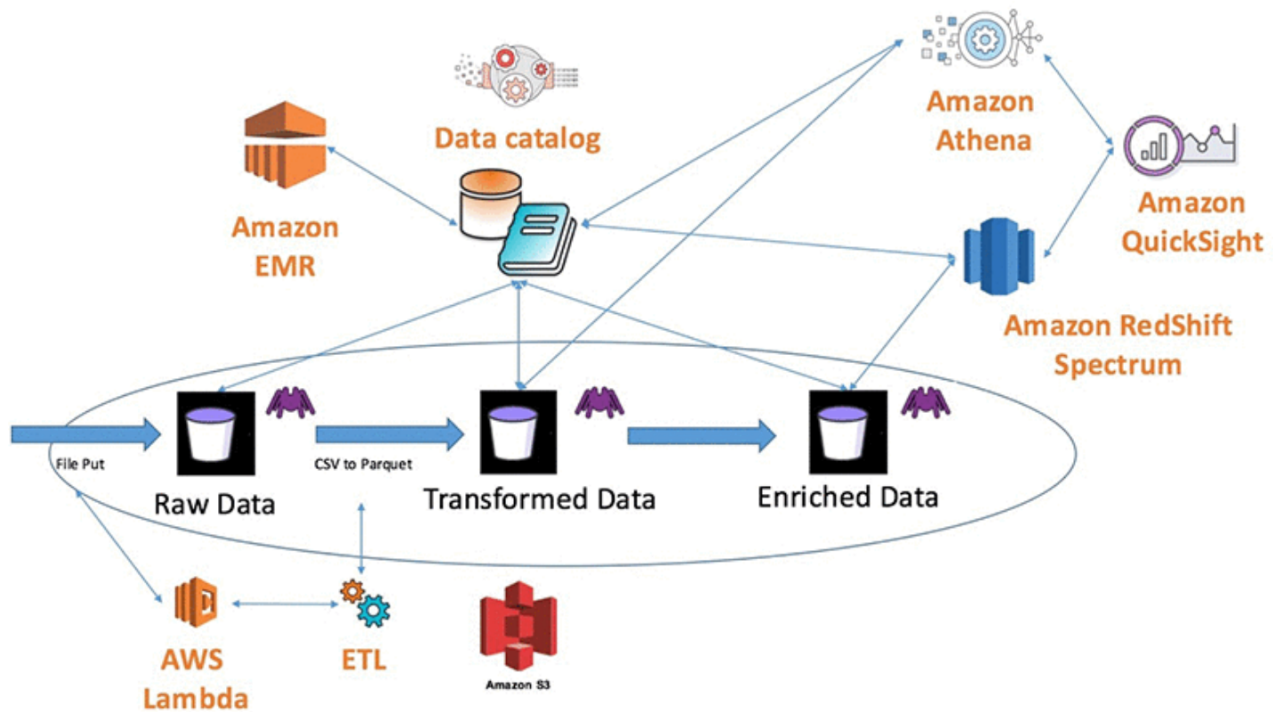
```

S3IOTest.UploadAndDownloadS3Throttle10
Tests passed: 1 of 1 test = 11 sec 111 ms
[WARN ] [2024-01-10 17:40:05,903] [main] [The deprecated 'path-style-access' property was used to specify access style. Please use 'access-style' instead.]
Try Upload
Wait for UploadCompleted
Upload complete: http://my-bucket.s3.localhost.localstack.cloud:4566/example.txt
Try Download
Download complete: Pair(Source(SourceShape(single.out(1109200052))), akka.stream.alpakka.s3.ObjectMetadata@e9831e2)

```

- reactive stream publisher/subscribe .
 - <https://github.com/psmon/java-labs/blob/master/springweb/src/test/java/com/webnori/springweb/alpakka/README.md> -
 - <https://github.com/psmon/java-labs/blob/master/springweb/src/test/java/com/webnori/springweb/alpakka/s3/S3IOTest.java> - s3
 - <https://github.com/psmon/java-labs/blob/master/springweb/src/test/java/com/webnori/springweb/alpakka/kafka/KafkaTest.java> - kafka
 - <https://github.com/psmon/java-labs/blob/master/springweb/src/test/java/com/webnori/springweb/webflux/BasicGuideTest.java> - webflux

- S3 Data



S3

- <https://aws.amazon.com/ko/blogs/korea/build-a-data-lake-foundation-with-aws-glue-and-amazon/>
 - S3 ~ Lake .
- <https://www.slideshare.net/Lightbend/lightbend-fast-data-platform>
 - 2017 Bigdata-DataWareHouse FastData S3 .
- <https://www.kai-waehner.de/blog/2022/06/27/data-warehouse-vs-data-lake-vs-data-streaming-friends-enemies-frenemies/>
 - DataLake ~ DataLake RealTime .

//

AWS-S3

OpenStack(Kafka/Spark/Nosql) .

~ OS First .

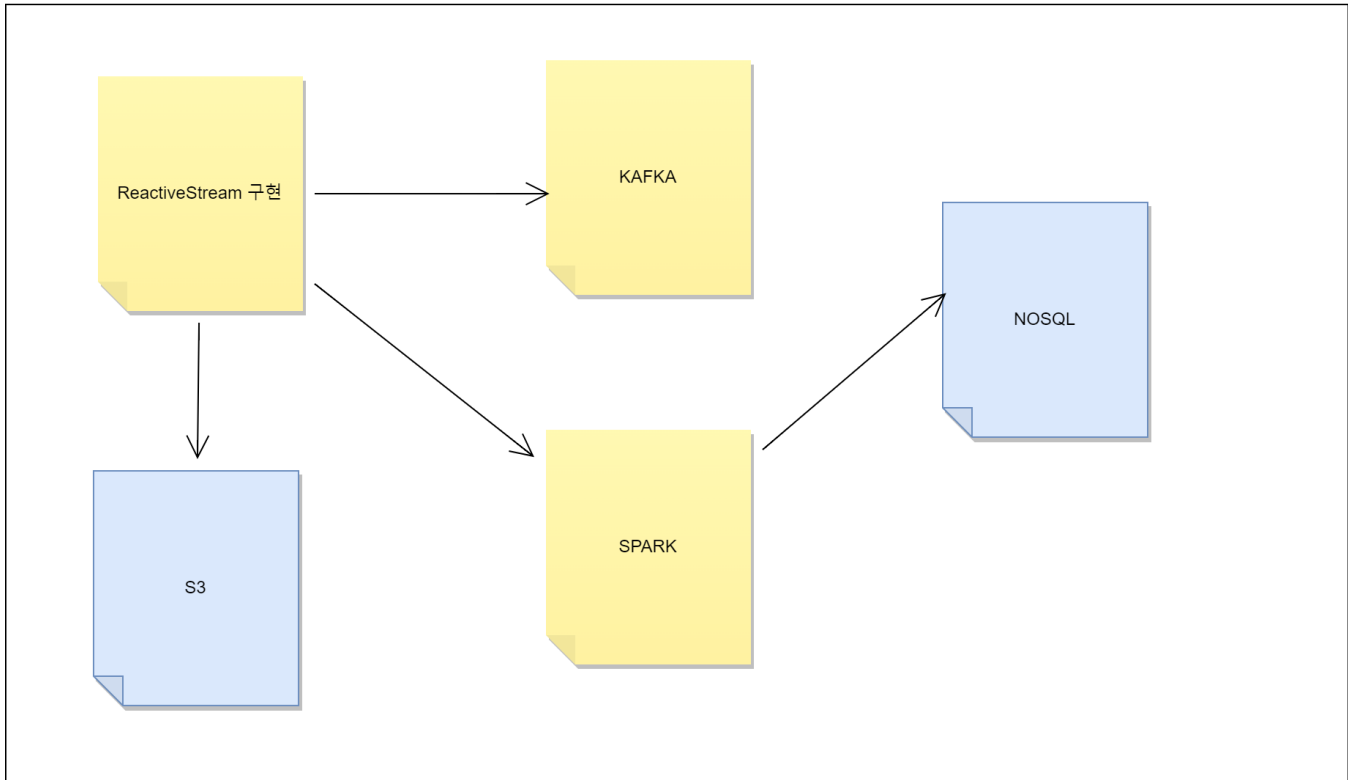
Local First~ .

first .

Aws .

- Athena - Spark
- S3 - LocalStack
- Redshift - Postgres
- Lambda - NoteBook Code
- OpenSearch - ElasticSearch

Local First



PaaS SE , .

100% ~ Local First .

- PaaS
- Pass
- AWS / Azure / GCP Pass
 - (ISP) .

OpenStack StandAlone .

- <https://github.com/psmon/java-labs/blob/master/infra/REAMME.md>

Next

- [AWS LocalStack - Lambda](#)
 - *Lambda* AWS , *LocalStack* .
 - *ETL* (Extract), (Transform), (Load) DataLake ETL .
- [Reactive Stream](#) - .