<epam>

Keeping **Secrets**

Istvan Zoltan Nagy

NOVEMBER 2024

Introduction

Istvan Zoltan Nagy
Solution Architect I

Experience

- 15+ years total
- 12+ years with EPAM

Tech stack

- Java,
- Spring, Spring Boot,
- AWS, Azure

Hobby

- Open-source developer

Contents

01

What are the secrets?

Let’s talk about them publicly!

“ a piece of information that is only known by one person or a few people and should not be told to others”

Cambridge Dictionary

For an IT engineer…

Company secrets

Some confidential information about your company of your clients.

Must be protected because we can get in trouble if leaked

Application secrets

Some confidential information needed for your application to function or be deployed

We shouldn’t know them in the first place

Why are application secrets so important?

Microsoft AI involuntarily exposed a secret giving access to 38TB of confidential data for 3 years
https://blog.gitguardian.com/microsoft-ai-involuntarily-exposed-a-secret-giving-access-to-38tb-of-confidential-data-for-3-years/

Misconfigurations in Google Firebase lead to over 19.8 million leaked secrets
https://blog.gitguardian.com/misconfigurations-in-google-firebase-lead-to-over-19-8-million-leaked-secrets/

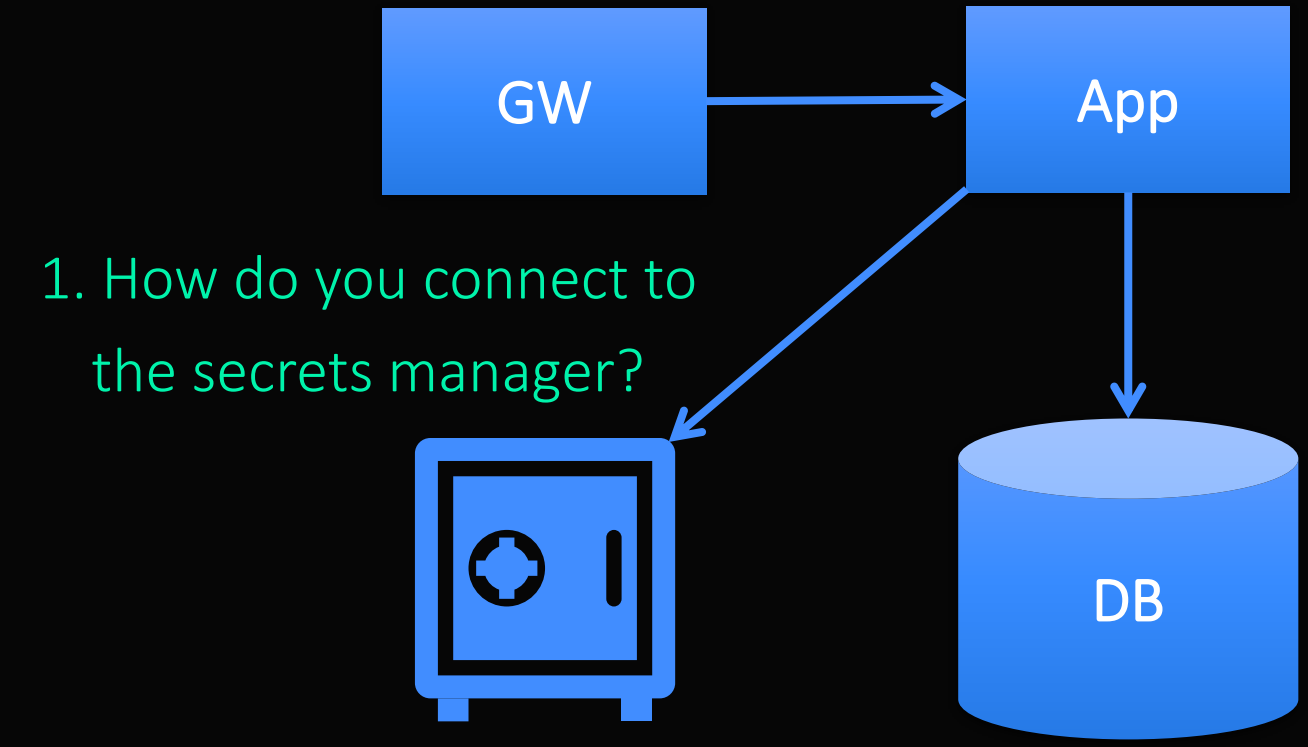
Nation-state hackers access Microsoft source code and steal secrets
https://blog.gitguardian.com/microsoft-breach-2024/

The Secrets of the New York Times Source Code Breach
https://blog.gitguardian.com/the-secrets-of-the-new-york-times-source-code-breach-2/

How Hackers Used Stolen GitHub Tokens to Access Private Source Code
https://blog.gitguardian.com/how-hackers-used-stolen-github-oauth-tokens/

An illustration of you and your application secrets

You
(“working”)



Using Managed Identity

System-assigned managed identity

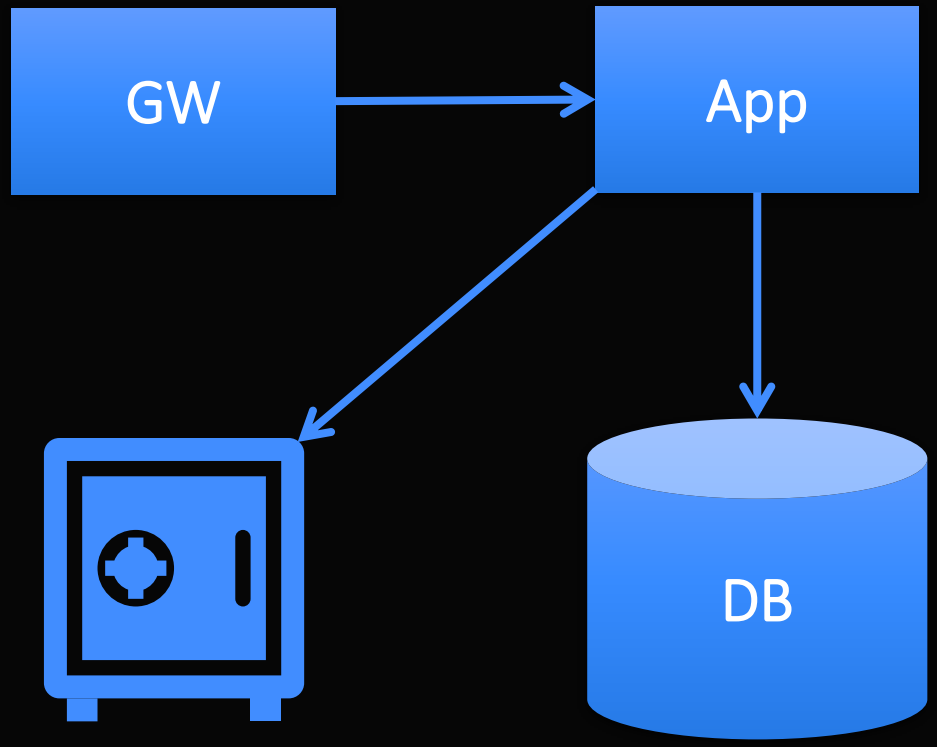
- Created with the resource
- Shared life cycle with the resource
- Only single resource

User-assigned managed identity

- Stand-alone resource
- Independent life cycle
- Can be shared

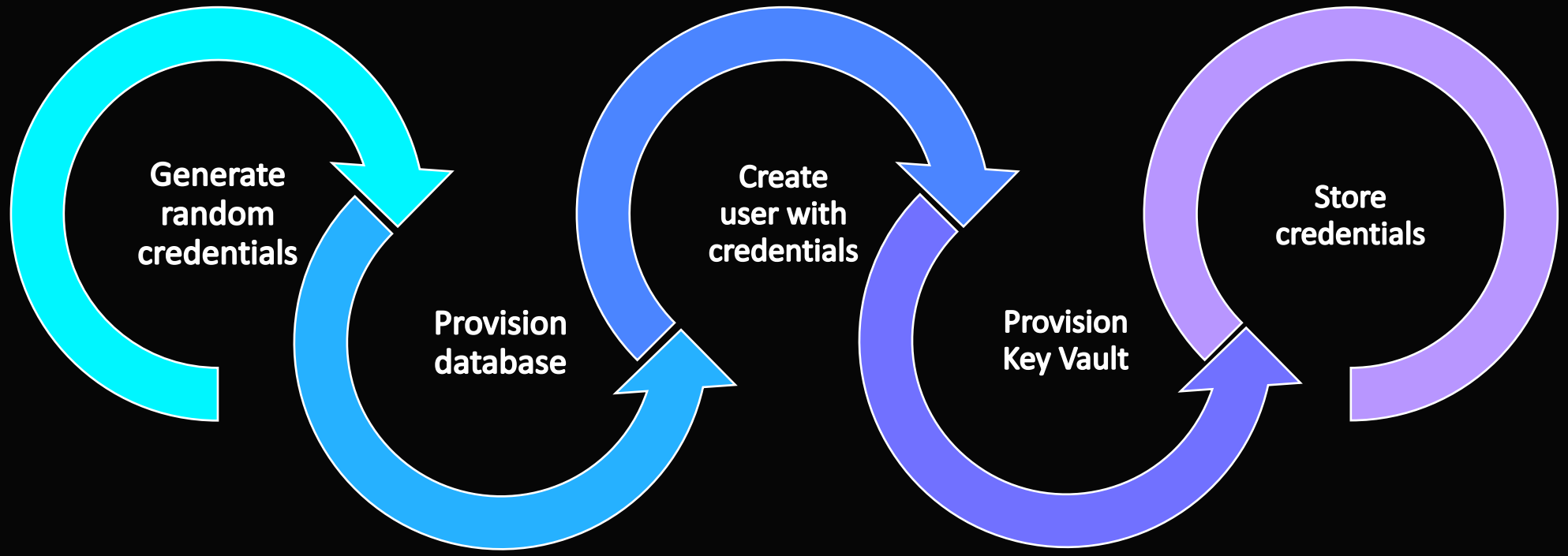
An illustration of you and your application secrets

You
(“working”)



2. How do you put the secrets there?

Setting your secrets from IaC



What does this mean for the developer?

Letting Kubernetes to resolve the secrets for us

Pro

- No impact, the application can be executed locally without changing anything

Con

- The secrets are resolved during start-up, you may need to restart the instance to pick up any change
- Supports only secrets and certificates

Resolving the secrets in our application

Pro

- Changes are easily noticed by the application runtime
- Supports key related use-cases as well

Con

- We have a new dependency that is needed to run the app locally

“ Wow, I need to connect to an *Azure Key Vault to start up my app locally! Best day ever! I hope I will need VPN too!”**

No developer ever

BUT…

***just an example, could be any cloud native secrets manager**

Let’s turn it over to the experts at Stack Overflow…

Question:

“Is it possible to have an [sic] local instance of Azure Key Vault?”

Answer:

“No, the reason is in the name: Azure Key Vault.
But you can setup your code to acquire credentials via a fallback mechanism when running in development mode, e.g. via user secrets.”

c# - Is it possible to have an local instance of Azure Key Vault? - Stack Overflow

What is wrong?

We need to

- either connect to a cloud service
- OR use some hack to circumvent the code accessing it

Which means that

- our tests are running on different code than Production
- we are shipping our code with some extra (test) code and dependencies in it
- we can test the real Azure Key Vault integration only late in the process

02

RUNNING IT LOCALLY

Let’s write a test double!

Just need to find out how…

Selecting the right kind of test double

Mock

- Setup (expectations)
- Use
- Verify behavior

Stub

- Provide canned responses
- Verify state

Fake object

- Working implementation
- Not suitable for production use

What did I implement for our fake object?

Functionality

Secrets

- 12 API endpoints supported

Keys

- 23 API endpoints supported
- Using RSA, EC, AES keys

Certificates

- 17 API endpoints supported
- Self-signed certificates only

Compatibility

Implemented API versions

- 7.2, 7.3, 7.4, 7.5

Supported clients

- Java
- JavaScript
- Go
- .Net
- Python

Ease of use

- No extra code or dependencies needed
- BYO HTTPS certificate
- Examples provided for each client
- Multiple vaults in the same container
- Import/export support
- (Fake) time travel

Proof of Concept Application

Spring Boot App

- Hello World
- Spring Cloud Azure Secrets Starter
- JDBC Starter

Azure Key Vault

- Real service on Prod
- Use our test double locally

MySQL Database

- Generic scenario
- Can be any other database

Extra Gradle configuration

```
bootRun {
    systemProperty("spring.profiles.active", "dev")
    systemProperty("javax.net.ssl.trustStore",
            file("$projectDir/local/lowkey-vault/lowkey-vault-keystore.p12"))
    systemProperty("javax.net.ssl.trustStorePassword", "changeit")
    systemProperty("javax.net.ssl.trustStoreType", "PKCS12")
    // Only needed if Assumed Identity and DefaultAzureCredential is used to
    // simulate IMDS managed identity
    environment "IDENTITY_ENDPOINT",
            "http://localhost:10544/metadata/identity/oauth2/token"
    environment "IDENTITY_HEADER", "header"
}
```

Extra App configuration needed (PROD)

```yaml
spring.cloud.azure.keyvault.secret:
  client.application-id: example
  property-source-enabled: true
  property-sources[0]:
    endpoint: http://real-azure-key-vault-endpoint-url
    refresh-interval: PT10S
    secret-keys:
    - spring-datasource-url
    - spring-datasource-driver-class-name
    - spring-datasource-username
    - spring-datasource-password
    service-version: V7_5
```

Extra App configuration needed (DEV)

```
# Use Lowkey Vault
spring.cloud.azure.keyvault.secret:
  property-sources[0]:
    challenge-resource-verification-enabled: false
    endpoint: https://localhost:10543/
```

Docker Compose config – Lowkey Vault

```yaml
services:
  lowkey-vault:
    container_name: spring-akv-example-lowkey-vault
    image: nagyesta/lowkey-vault:2.4.109
    ports:
      - "10543:10543"
      - "10544:10544"
    volumes:
      - ./lowkey-vault/data/keyvault.json.hbs:/data/keyvault.json.hbs
    environment:
      LOWKEY_ARGS: "--server.port=10543 --app.token.port=10544
        --LOWKEY_VAULT_NAMES=- --LOWKEY_IMPORT_LOCATION=/data/keyvault.json.hbs"
```

Docker Compose config - MySQL

```yaml
mysql:
  container_name: spring-akv-example-mysql
  image: mysql:8.0.39
  command: --default-authentication-plugin=mysql_native_password
  environment:
    MYSQL_ROOT_PASSWORD: NOT_A_SECRET_5b8538b6-2bf1-4d38-94f0-308d4fbb757b
  ports:
    - '23306:3306'
```

Lowkey Vault import file


```
{"vaults":[{"attributes":{"baseUri":"https://{{host}}:{{port}}",
"recoveryLevel":"Recoverable+Purgeable","recoverableDays":90,"created":{{now 0}},"deleted":null},
"keys":{},"secrets":{"spring-datasource-url":{"versions":[{"vaultBaseUri":"https://{{host}}:{{port}}",
"entityId":"spring-datasource-url","entityVersion":"00000000000000000000000000000001",
"attributes":{"enabled":true,"created":{{now 0}},"updated":{{now 0}},
"recoveryLevel":"Recoverable+Purgeable","recoverableDays":90},"tags":{},"managed":false,
"value":"jdbc:mysql://localhost:23306/","contentType":"text/plain"}]},
"spring-datasource-username":{"versions":[{"vaultBaseUri":"https://{{host}}:{{port}}",
"entityId":"spring-datasource-username","entityVersion":"00000000000000000000000000000001",
"attributes":{"enabled":true,"created":{{now 0}},"updated":{{now 0}},
"recoveryLevel":"Recoverable+Purgeable","recoverableDays":90},"tags":{},"managed":false,"value":"root",
"contentType":"text/plain"}]},"spring-datasource-password":{"versions":[{"vaultBaseUri":"https://{{host
}}:{{port}}","entityId":"spring-datasource-password",
"entityVersion":"00000000000000000000000000000001","attributes":{"enabled":true,"created":{{now
0}},"updated":{{now 0}},"recoveryLevel":"Recoverable+Purgeable","recoverableDays":90},"tags":{},
"managed":false,"value":"NOT_A_SECRET_5b8538b6-2bf1-4d38-94f0-308d4fbb757b",
"contentType":"text/plain"}]},"spring-datasource-driver-class-name":{"versions":[{"vaultBaseUri":"https
://{{host}}:{{port}}","entityId":"spring-datasource-driver-class-name",
"entityVersion":"00000000000000000000000000000001","attributes":{"enabled":true,"created":{{now
0}},"updated":{{now 0}},"recoveryLevel":"Recoverable+Purgeable","recoverableDays":90},"tags":{},
"managed":false,"value":"com.mysql.cj.jdbc.Driver","contentType":"text/plain"}]}}}]}
```

Lowkey Vault import file (partial)

1 of 4 secrets

- URL
- Driver class name
- Username
- Password

Local credentials

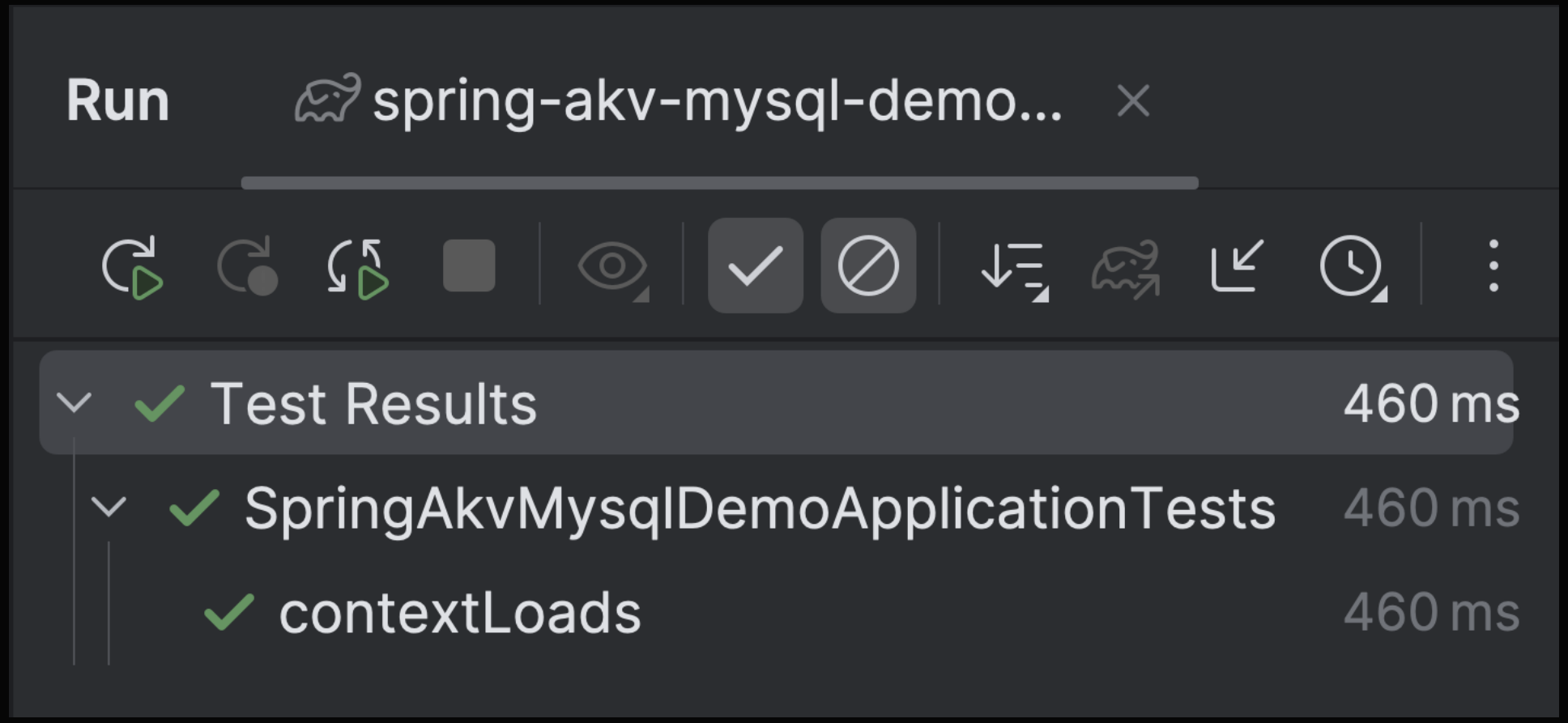
- Throw-away DB
- No need to protect them

Time-travel

- Created and updated times are relative to current timestamp


```
"secrets": {
  "spring-datasource-url": {
    "versions": [
      {
        "vaultBaseUri": "https://{{host}}:{{port}}",
        "entityId": "spring-datasource-url",
        "entityVersion": "00000000000000000000000000000001",
        "attributes": {
          "enabled": true,
          "created": {{now 0}},
          "updated": {{now 0}},
          "recoveryLevel": "Recoverable+Purgeable",
          "recoverableDays": 90
        },
        "tags": {},
        "managed": false,
        "value": "jdbc:mysql://localhost:23306/",
        "contentType": "text/plain"
      }
    ]
  },
```

Does it work?


Run
spring-akv-mysql-demo...
Test Results
460 ms
SpringAkvMysqlDemoApplicationTests
460 ms
contextLoads
460 ms

What did we learn?

Summary

- We must test what we will use in Production
- We can, and should, do better when we handle application secrets
- Depending on Azure Key Vault locally is not the end of the World
  - or AWS Secret Manager
  - or HashiCorp Vault

References

- Lowkey Vault Project home - https://github.com/nagyesta/lowkey-vault
- Example project - https://github.com/nagyesta/spring-akv-mysql-demo
- GitGuardian - https://www.gitguardian.com
- Azure Key Vault - https://azure.microsoft.com/en-us/products/key-vault
- Entra ID Managed Identities - https://learn.microsoft.com/en-us/entra/identity/managed-identities-azure-resources/overview
- Spring Boot - https://spring.io/projects/spring-boot
- Mocks vs Stubs - https://martinfowler.com/articles/mocksArentStubs.html

Do you have any questions?

Thank you!

For more information, contact

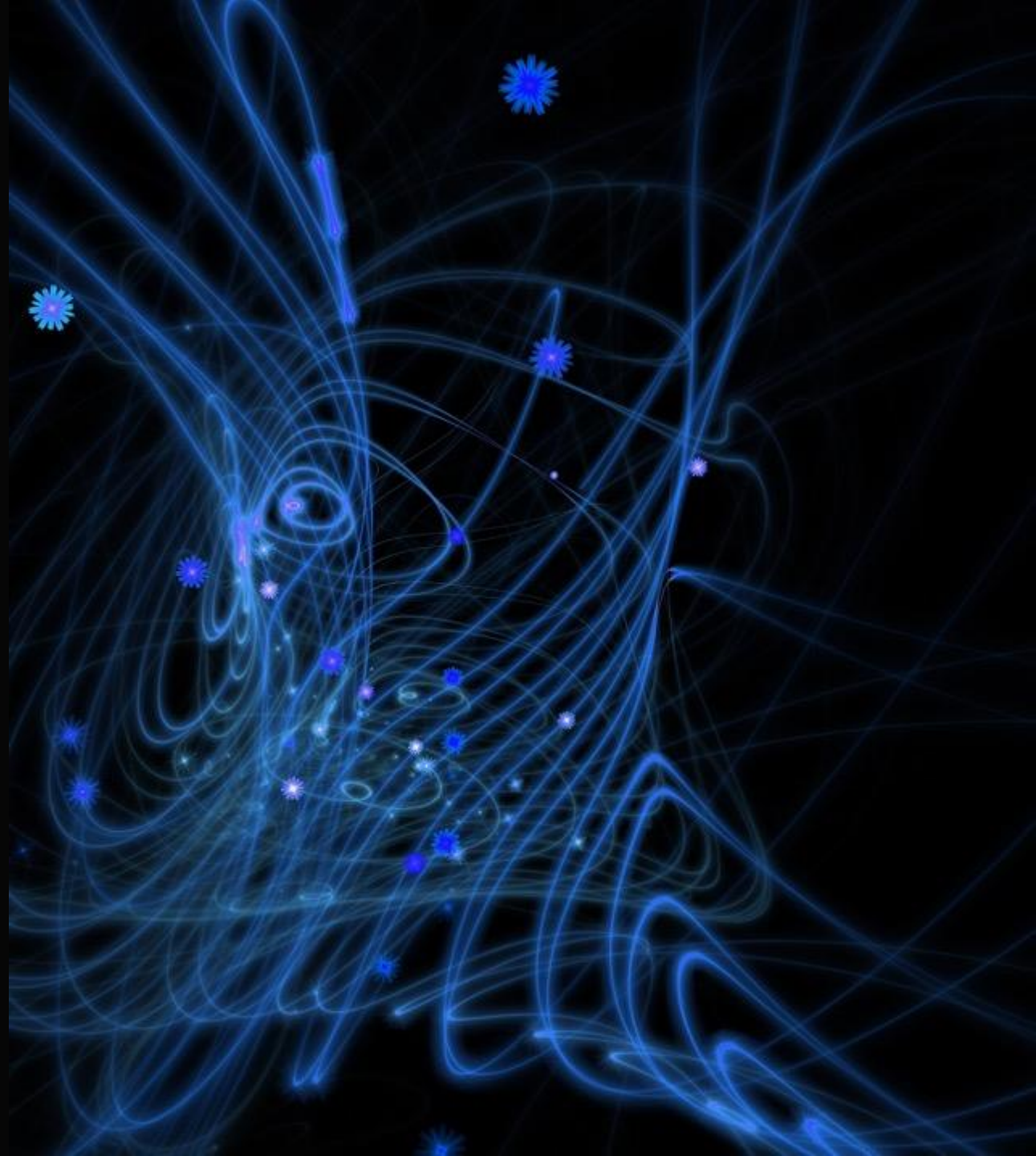
Istvan Zoltan Nagy

Solution Architect I

Email: Istvan_Nagy@epam.com

GitHub: nagyesta

LinkedIn: istvan-zoltan-nagy-b0a42b1b4



‹epam›