Are Your Organization's CI/CD Pipelines Secure?

Angel Rivera - @punkdata

Developer Advocate

circleci

DEVOPS WORLD by CloudBees

AGENDA

- Intro
- SDLC + CI/CD
- DevSecOps + Pipeline Integrations
- Securing Pipelines
- Recap

@punkdata

Hi I'm Angel

Developer Advocate.

My job is to **inspire** developers!

@punkdata

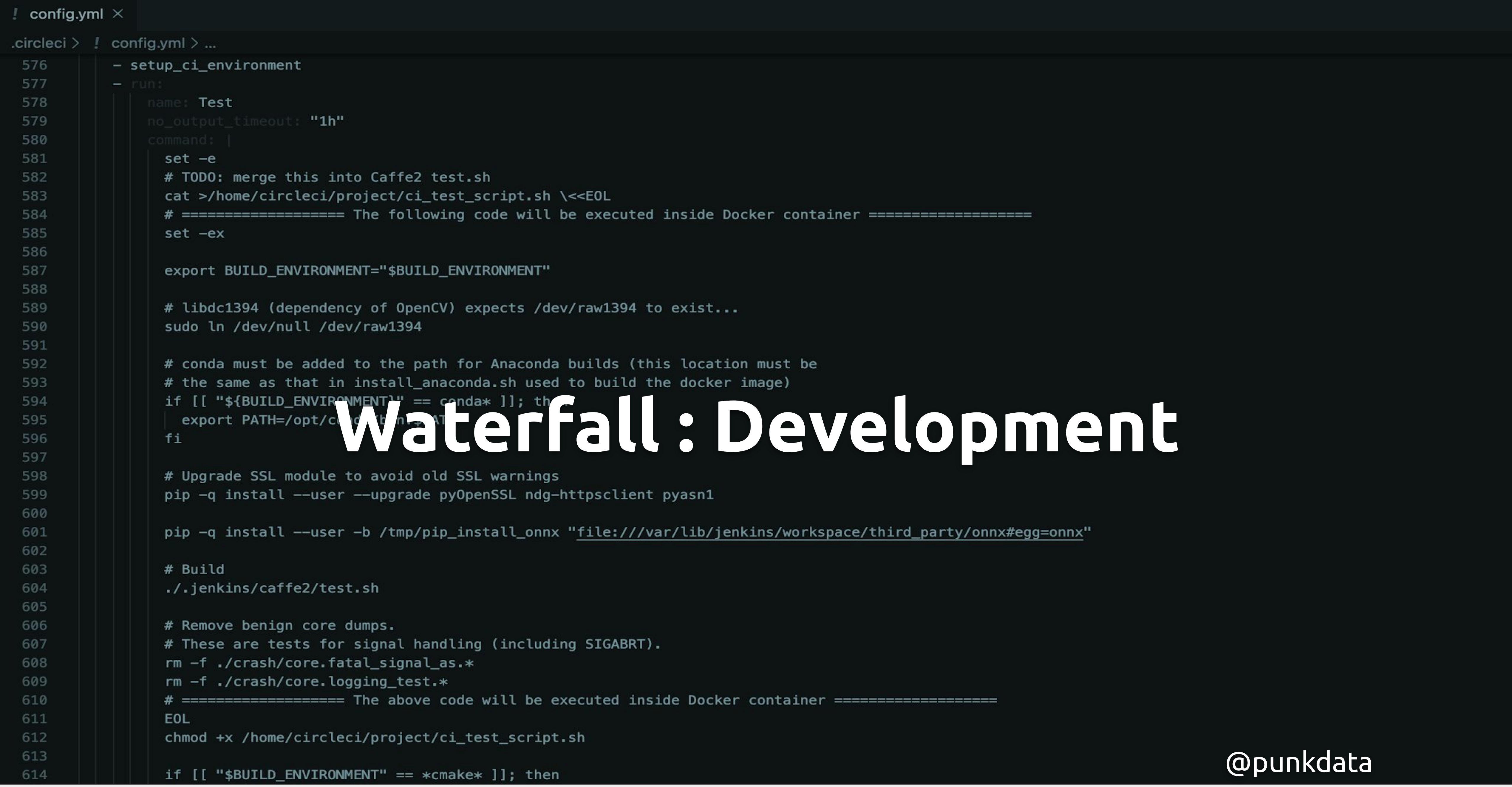
History : Software : Development

@punkdata

Software : Development : Practices

@punkdata

Waterfall : Development

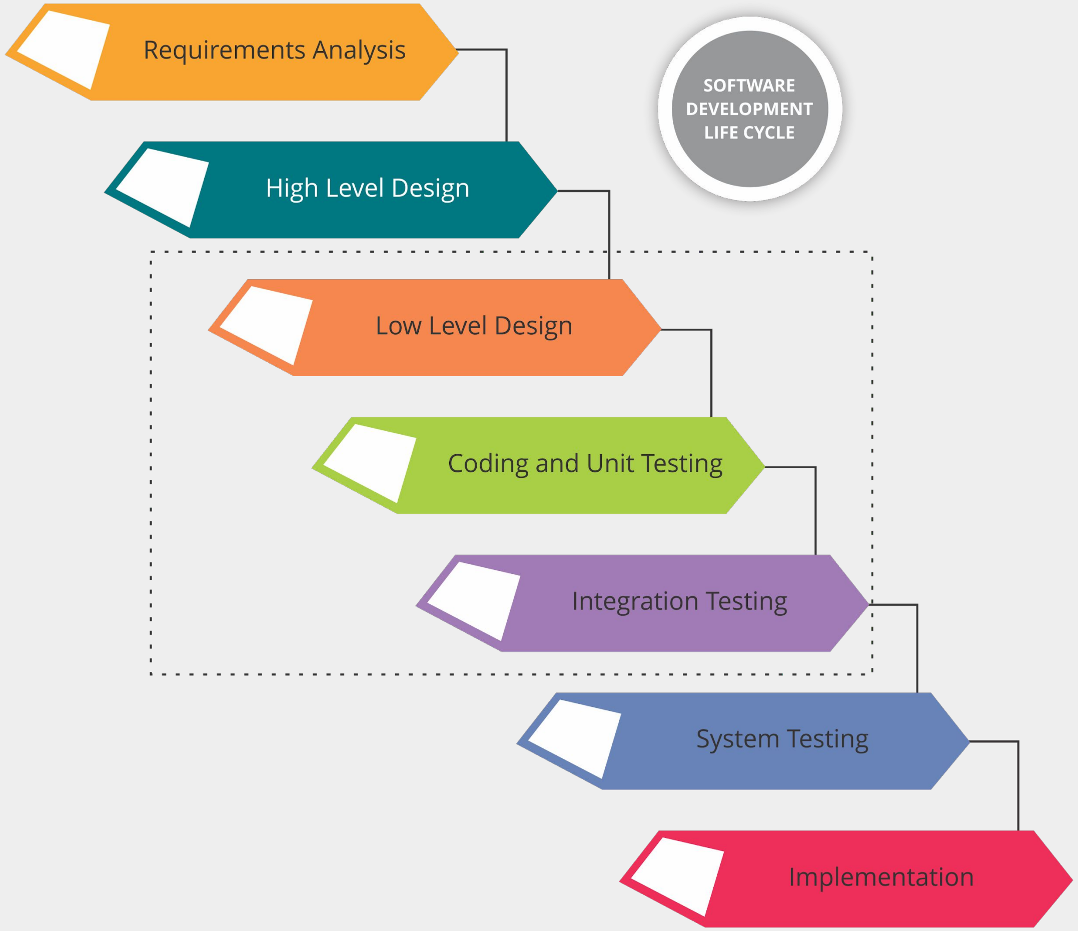


```
- setup_ci_environment
- run:
    name: Test
    no_output_timeout: "1h"
    command: |
      set -e
      # TODO: merge this into Caffe2 test.sh
      cat >/home/circleci/project/ci_test_script.sh \<<EOL
      # =================== The following code will be executed inside Docker container ===================
      set -ex

      export BUILD_ENVIRONMENT="$BUILD_ENVIRONMENT"

      # libdc1394 (dependency of OpenCV) expects /dev/raw1394 to exist...
      sudo ln /dev/null /dev/raw1394

      # conda must be added to the path for Anaconda builds (this location must be
      # the same as that in install_anaconda.sh used to build the docker image)
      if [[ "${BUILD_ENVIRONMENT}" == conda* ]]; then
        export PATH=/opt/conda/bin:$PATH
      fi

      # Upgrade SSL module to avoid old SSL warnings
      pip -q install --user --upgrade pyOpenSSL ndg-httpsclient pyasn1

      pip -q install --user -b /tmp/pip_install_onnx "file:///var/lib/jenkins/workspace/third_party/onnx#egg=onnx"

      # Build
      ./.jenkins/caffe2/test.sh

      # Remove benign core dumps.
      # These are tests for signal handling (including SIGABRT).
      rm -f ./crash/core.fatal_signal_as.*
      rm -f ./crash/core.logging_test.*
      # =================== The above code will be executed inside Docker container ===================
      EOL
      chmod +x /home/circleci/project/ci_test_script.sh

      if [[ "$BUILD_ENVIRONMENT" == *cmake* ]]; then
```

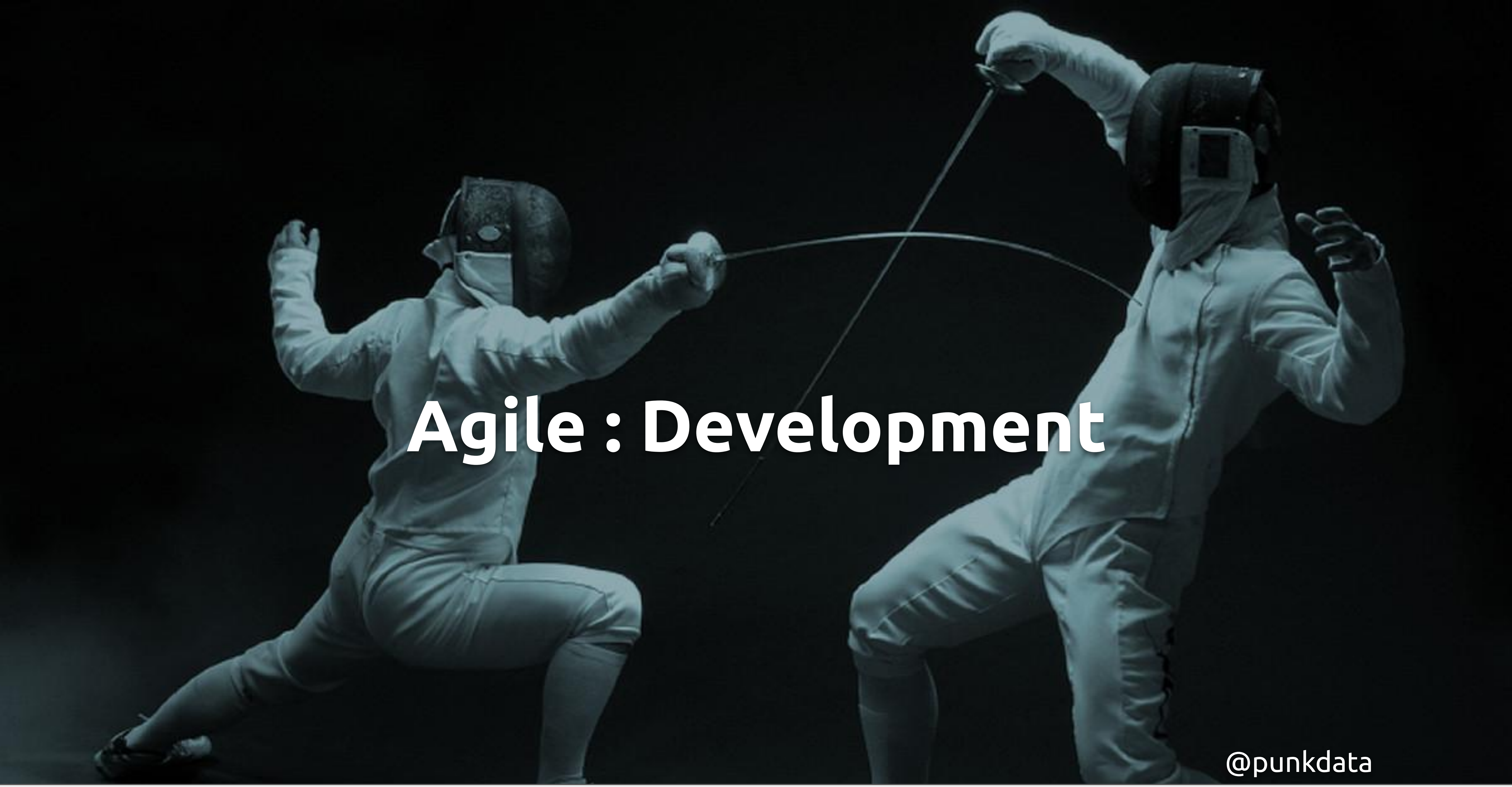
@punkdata

DEVOPSWORLD
by CloudBees

Waterfall : Development



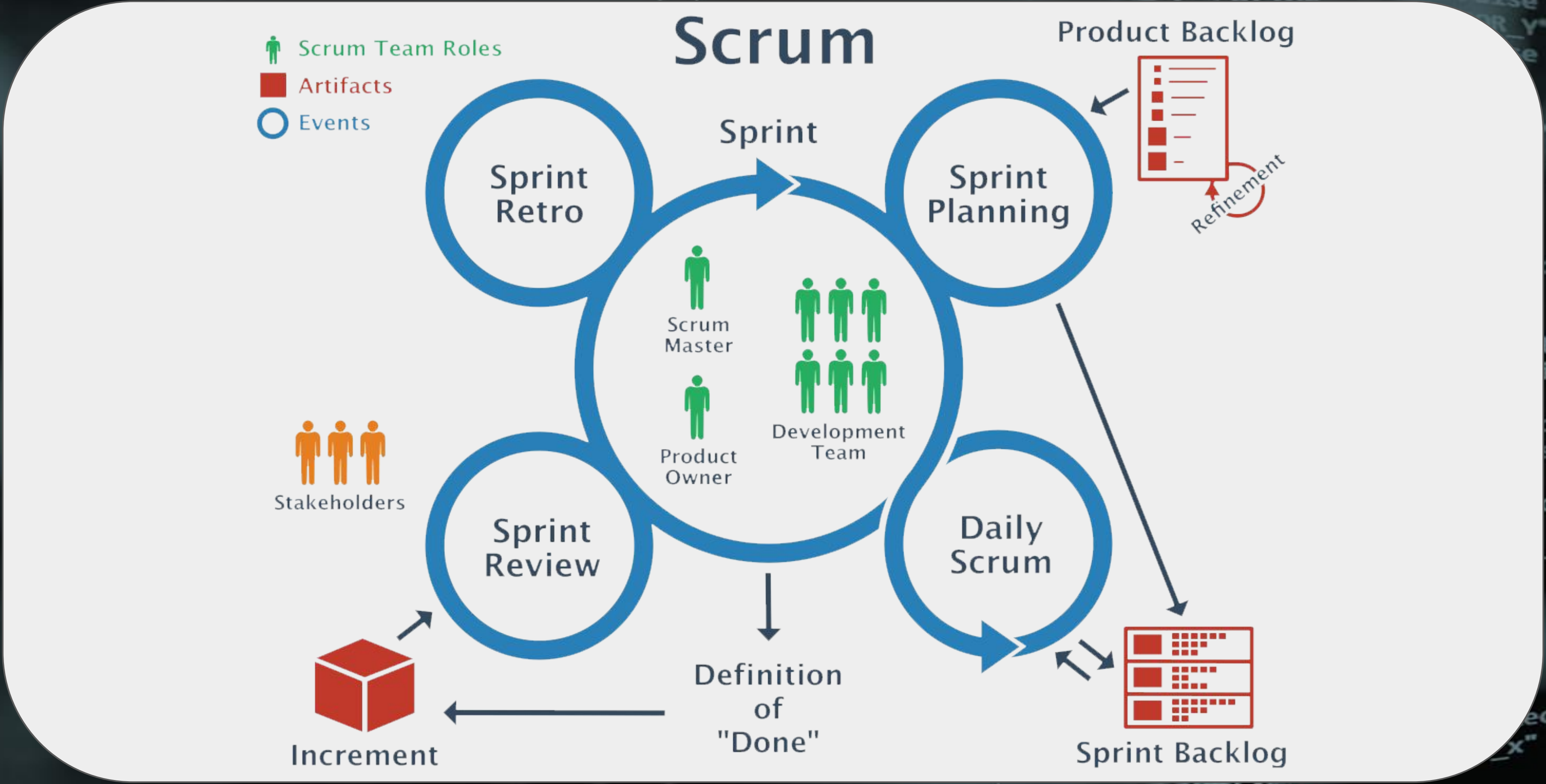
@punkdata



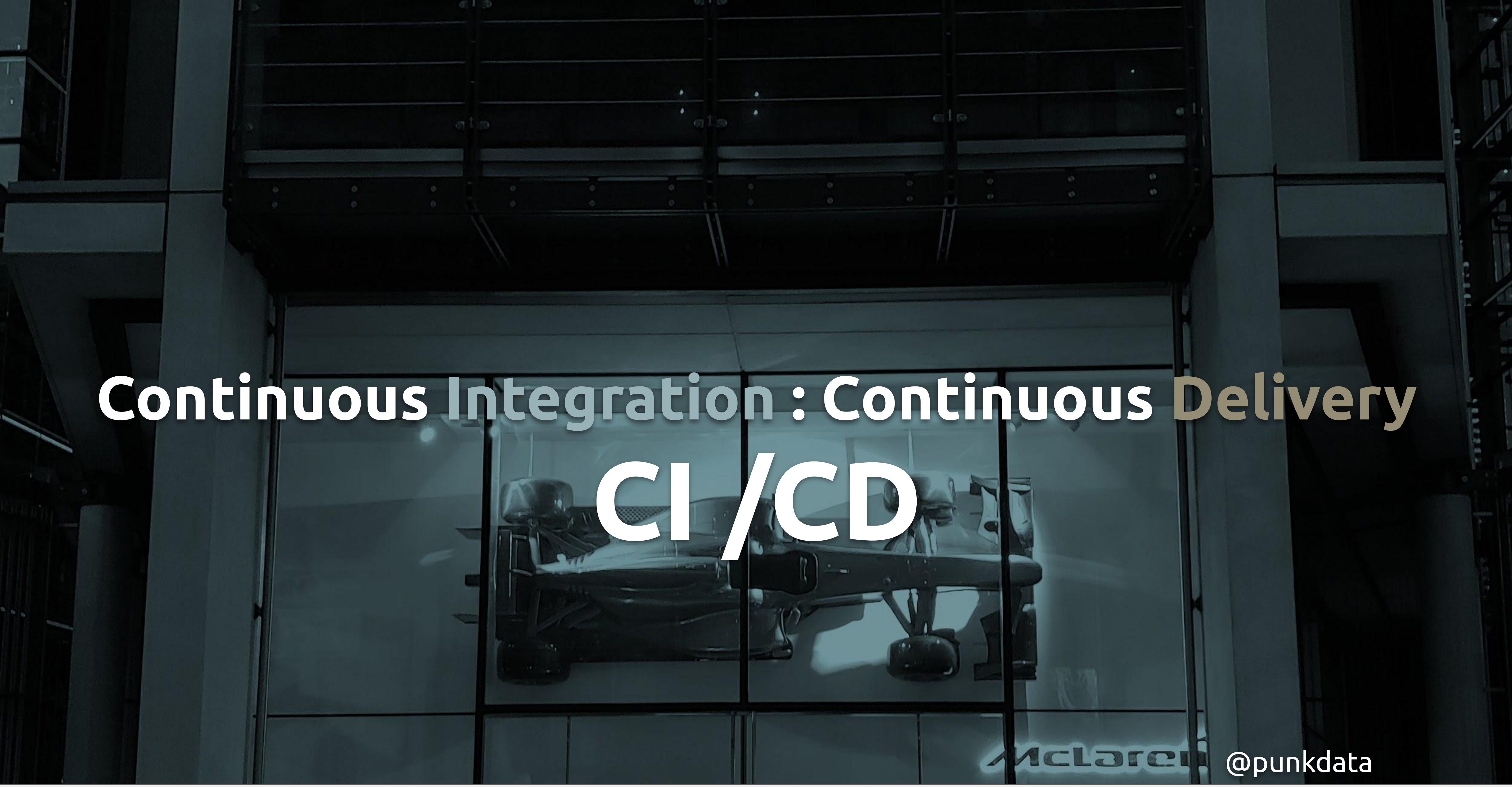
Agile : Development

@punkdata

Agile : Development



@punkdata

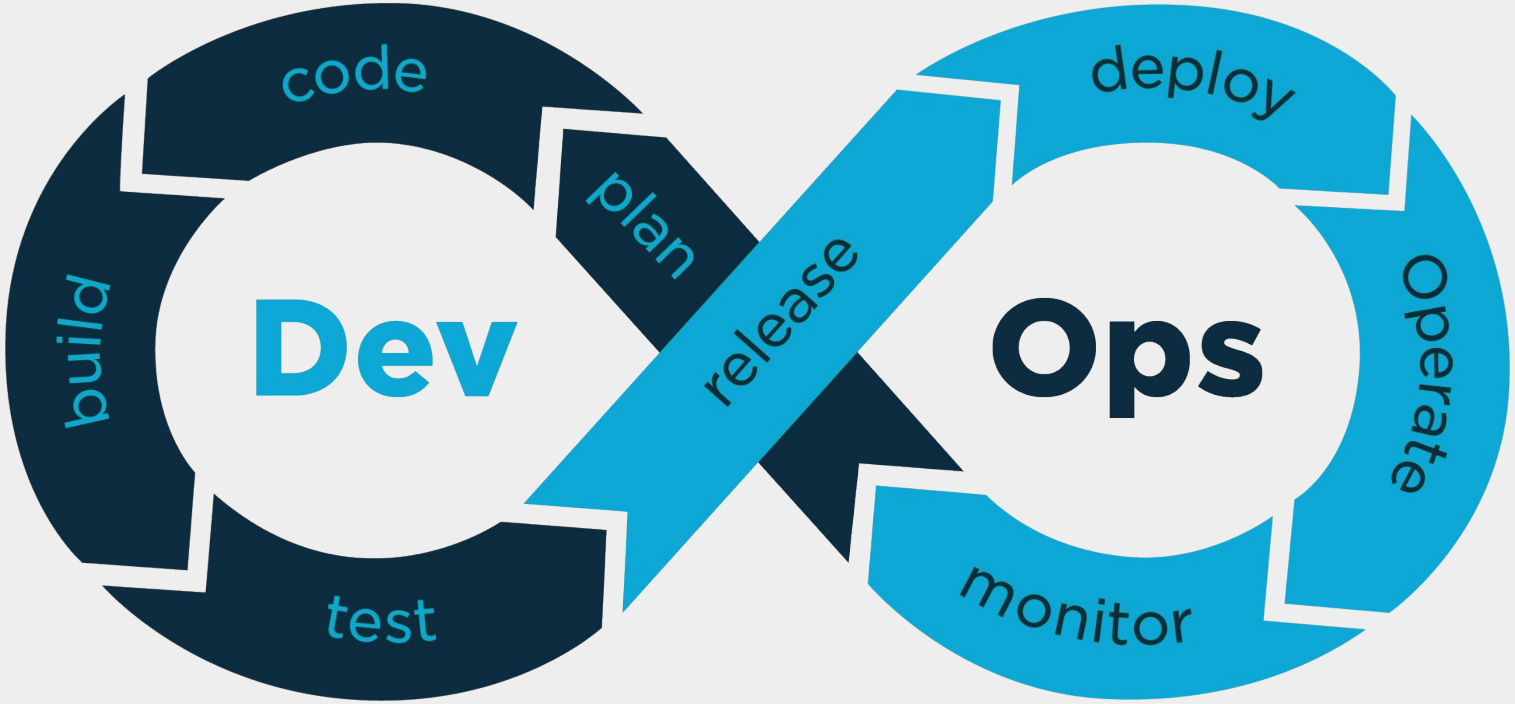


Continuous Integration : Continuous Delivery

CI /CD

@punkdata

CI/CD + DevOps



@punkdata

CI : Dev : Principles

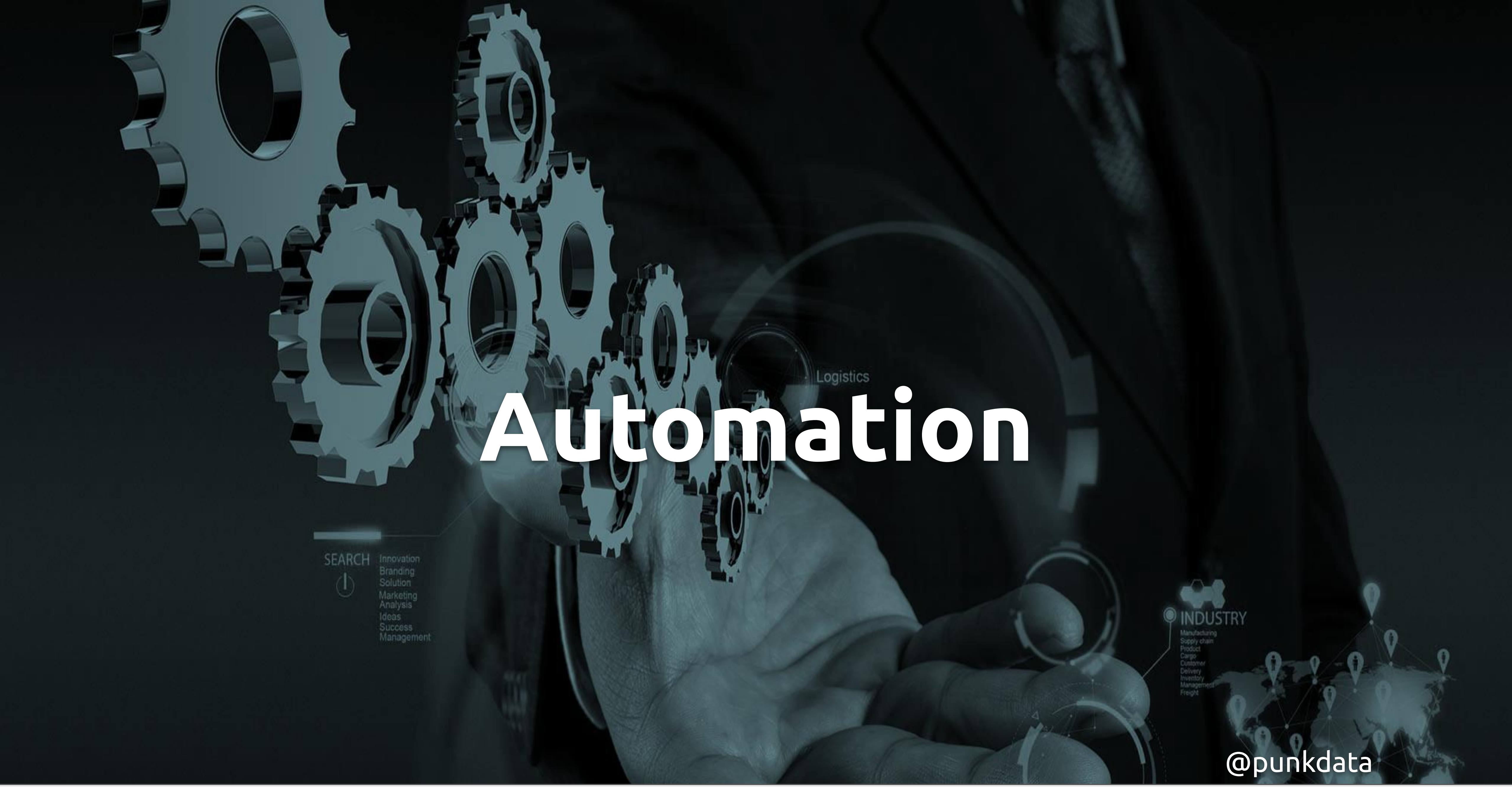
- Write and commit code often
- Commit to a shared code repository
- Test code on every commit (unit + smoke level tests)
- Fast Feedback Loops

@punkdata

CD : Ops : Principles

- Create release artifacts for CD process
- Deploy code artifacts to resources
- Validate apps and services are functioning
- Monitor to verify state and recover if failing

@punkdata



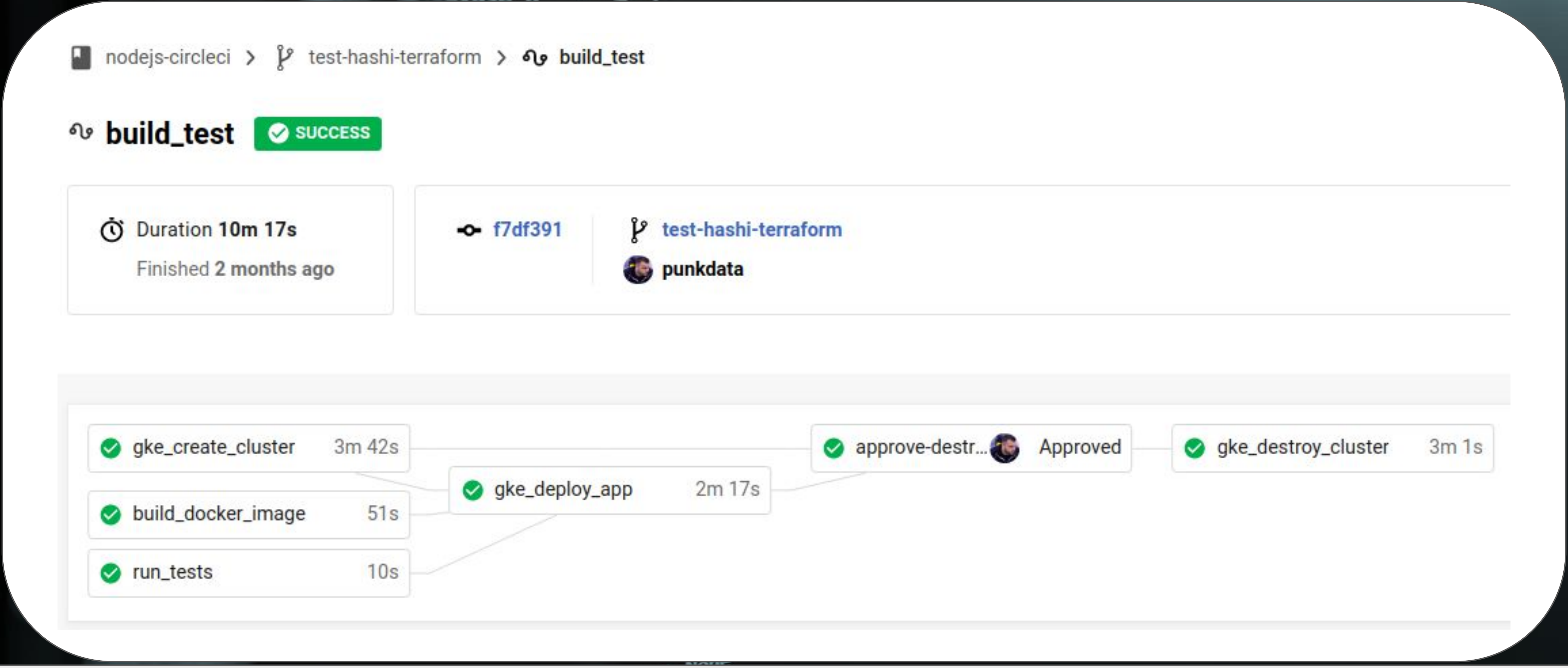
Automation

@punkdata

CI/CD : Pipelines

@punkdata

CI/CD : Pipelines



nodejs-circleci > test-hashi-terraform > build_test

build_test ✓ SUCCESS

Duration **10m 17s**

Finished **2 months ago**

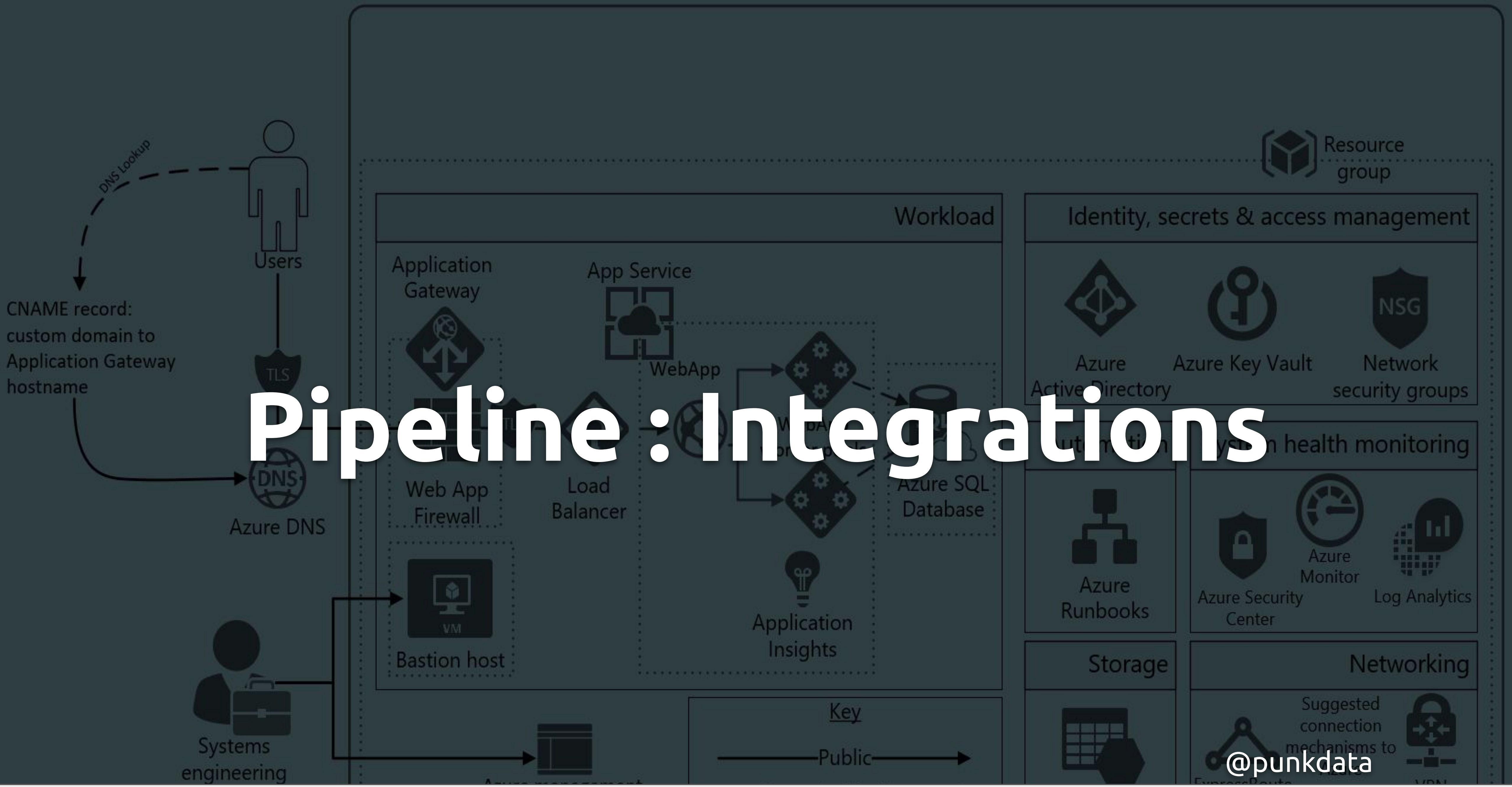
f7df391

test-hashi-terraform

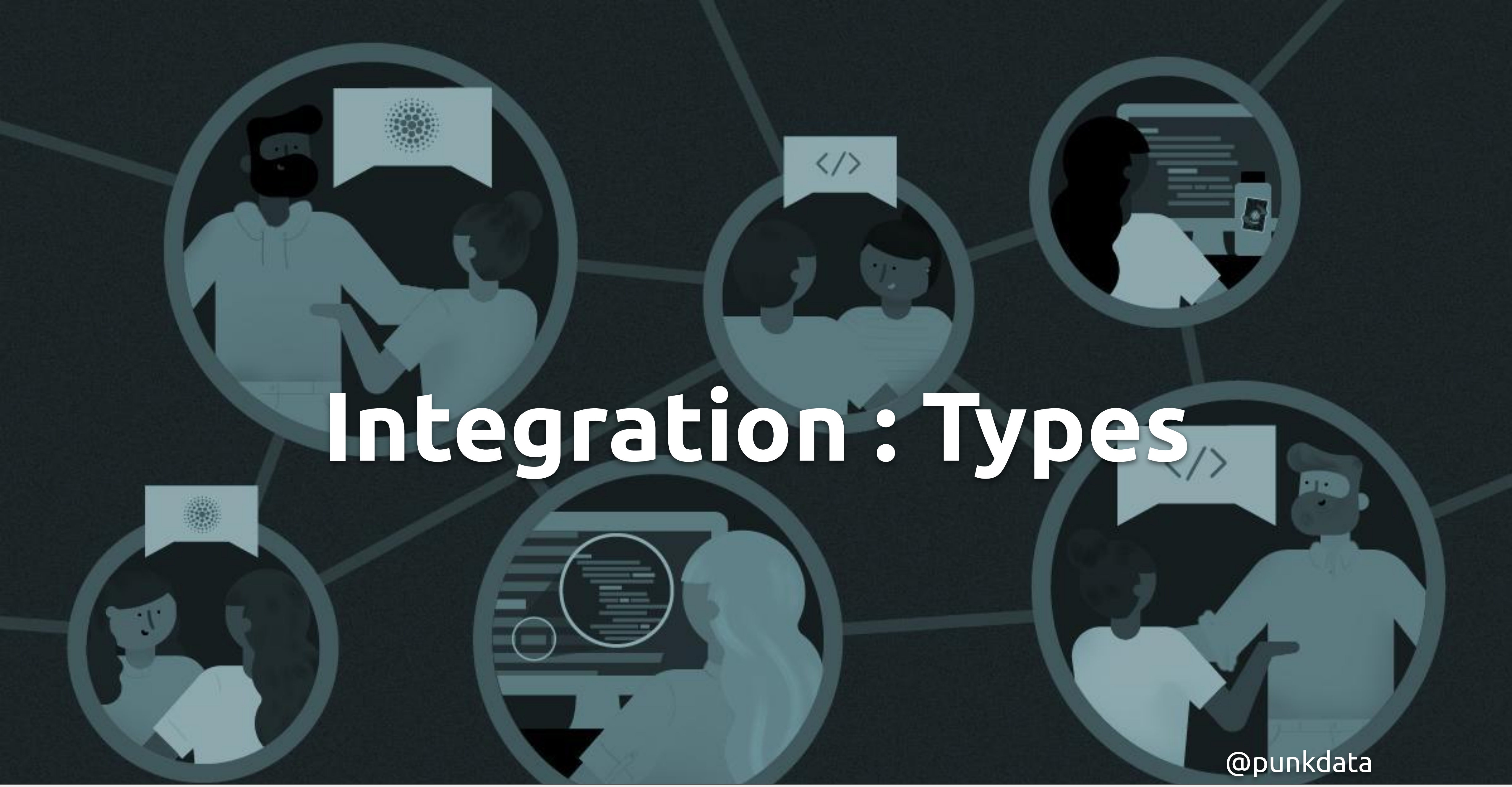
punkdata

✓ gke_create_cluster	3m 42s
✓ build_docker_image	51s
✓ run_tests	10s
✓ gke_deploy_app	2m 17s
✓ approve-destr...	Approved
✓ gke_destroy_cluster	3m 1s

Pipeline : Integrations



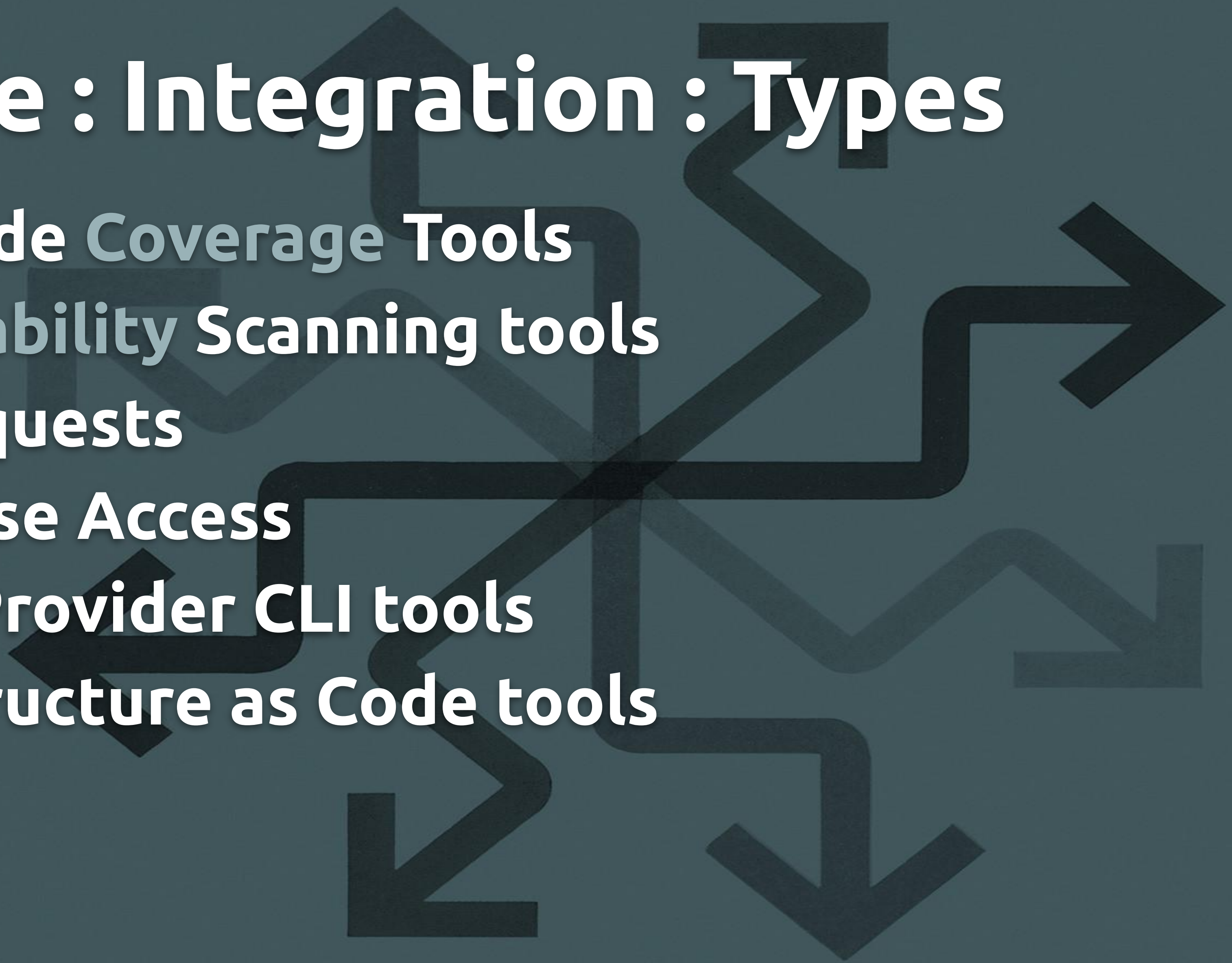
@punkdata



Integration : Types

@punkdata

Pipeline : Integration : Types



- App Code Coverage Tools
- Vulnerability Scanning tools
- API Requests
- Database Access
- Cloud Provider CLI tools
- Infrastructure as Code tools

@punkdata

Dev : Sec : Ops

@punkdata

Shift : Security : Left

@punkdata

Pipeline : Vulnerability : Scans

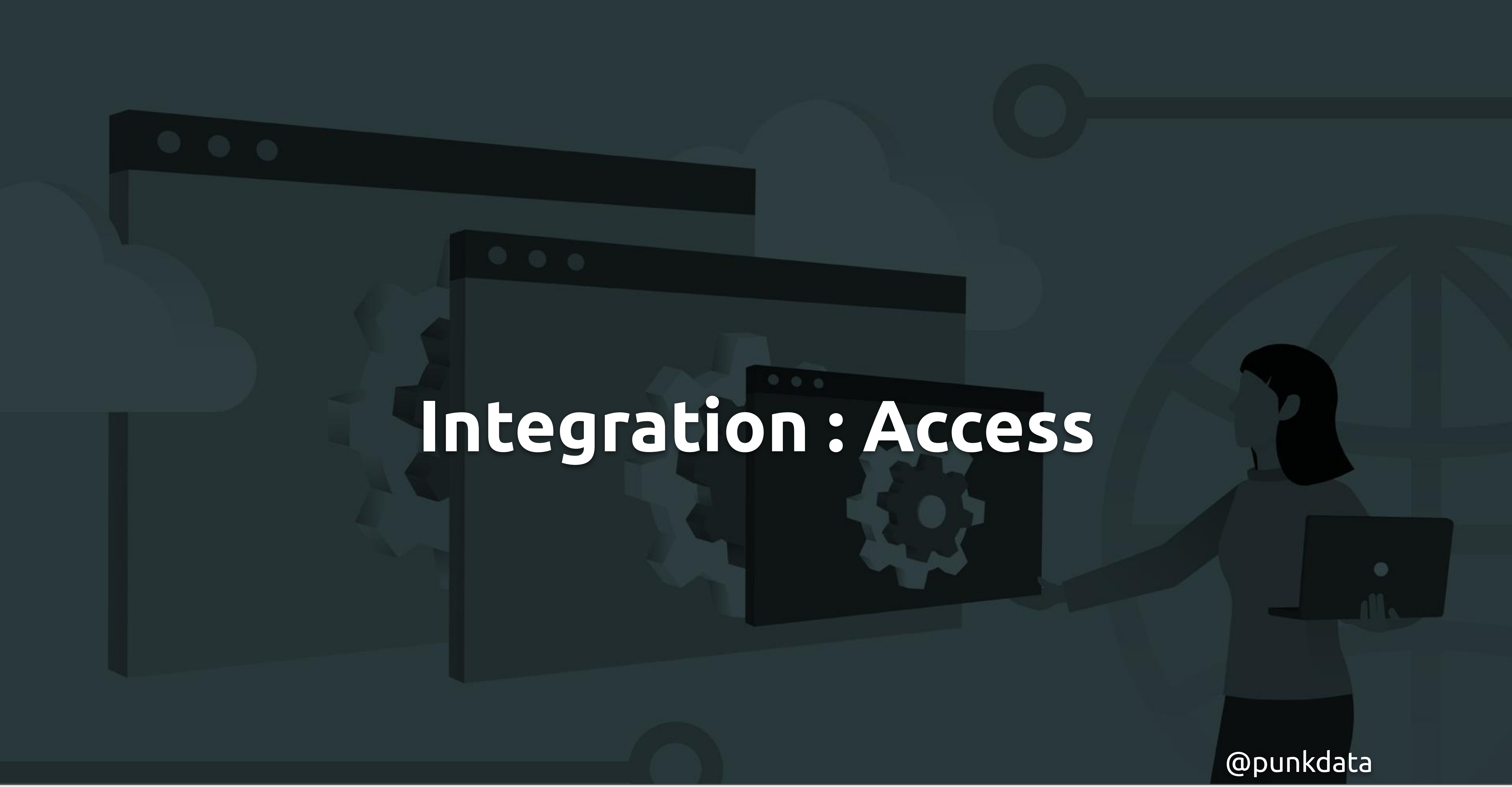
@punkdata

App : Scanning : Pipelines

@punkdata

Container : Scanning : Pipelines

@punkdata



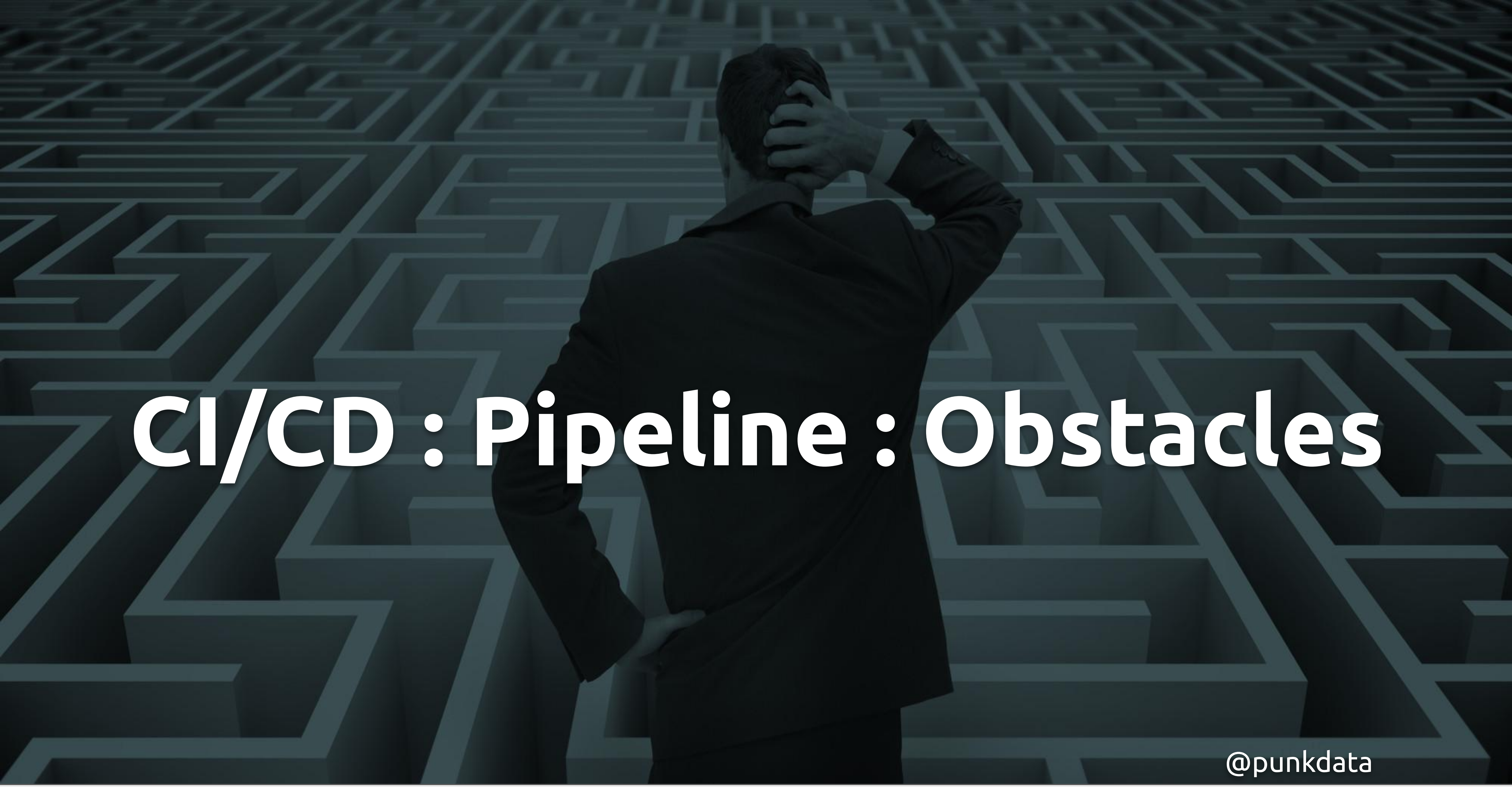
Integration : Access

@punkdata

Integration : Access : Credentials

- OAuth
- Access Keys
- API Tokens
- Certificate-based authentication
- JWT - JSON Web Tokens
- Username + Password

@punkdata



CI/CD : Pipeline : Obstacles

@punkdata

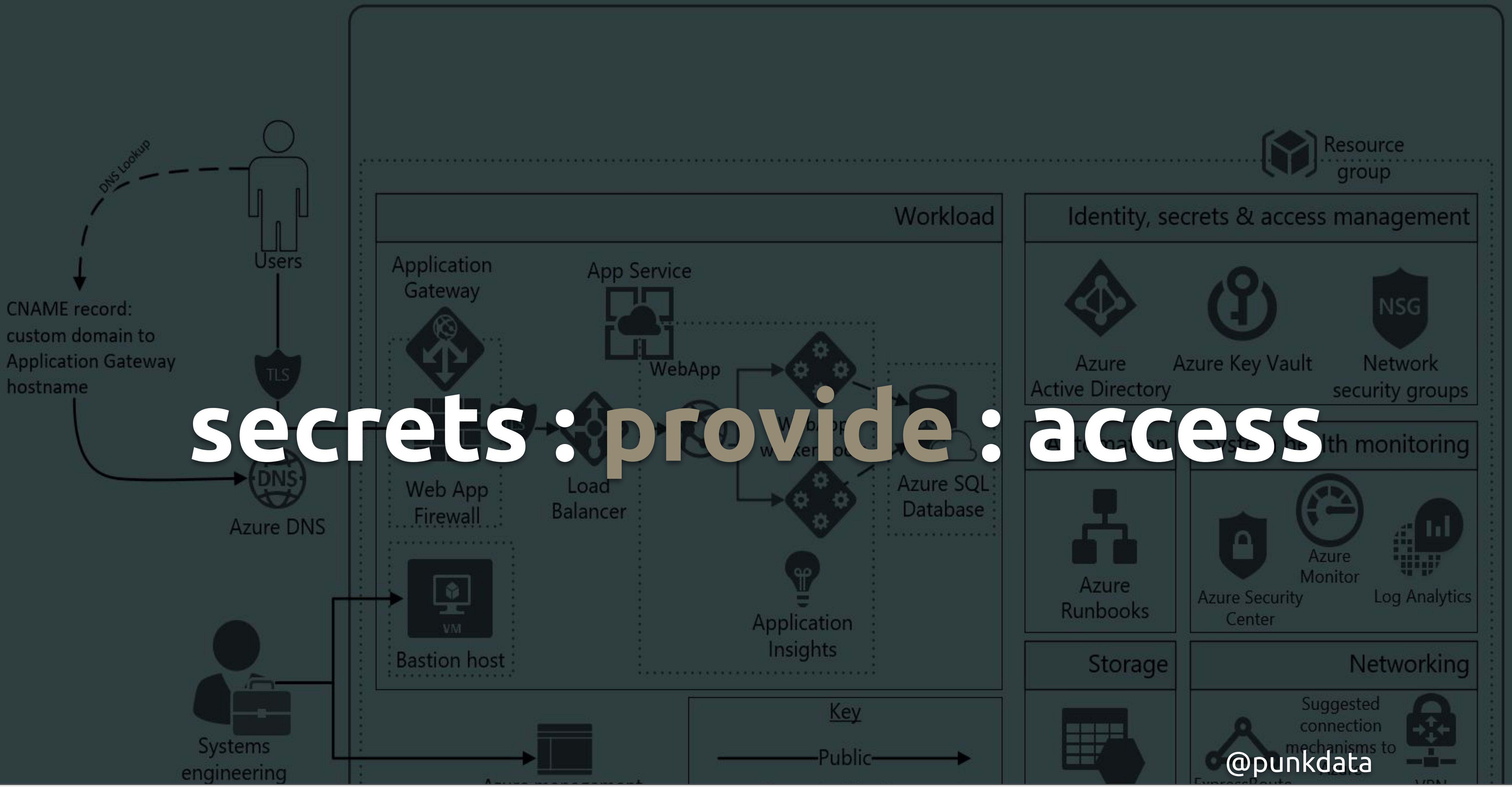
Securing Pipelines

@punkdata

Secrets

@punkdata

secrets : provide : access



@punkdata

exposed : pipelines

@punkdata

weak : stale : secrets

@punkdata

clear text

inadequate : mechanisms

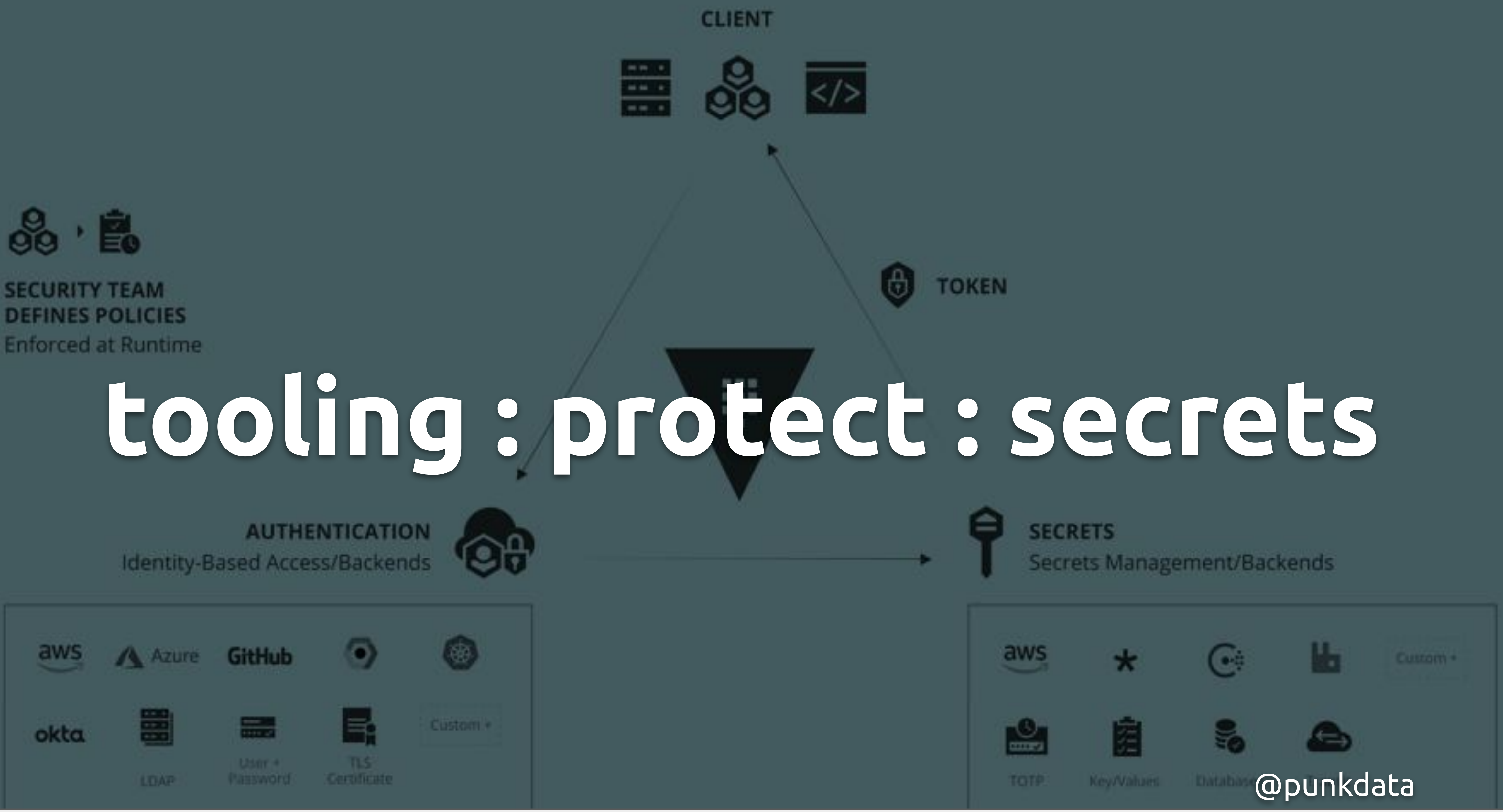
unencrypted

@punkdata

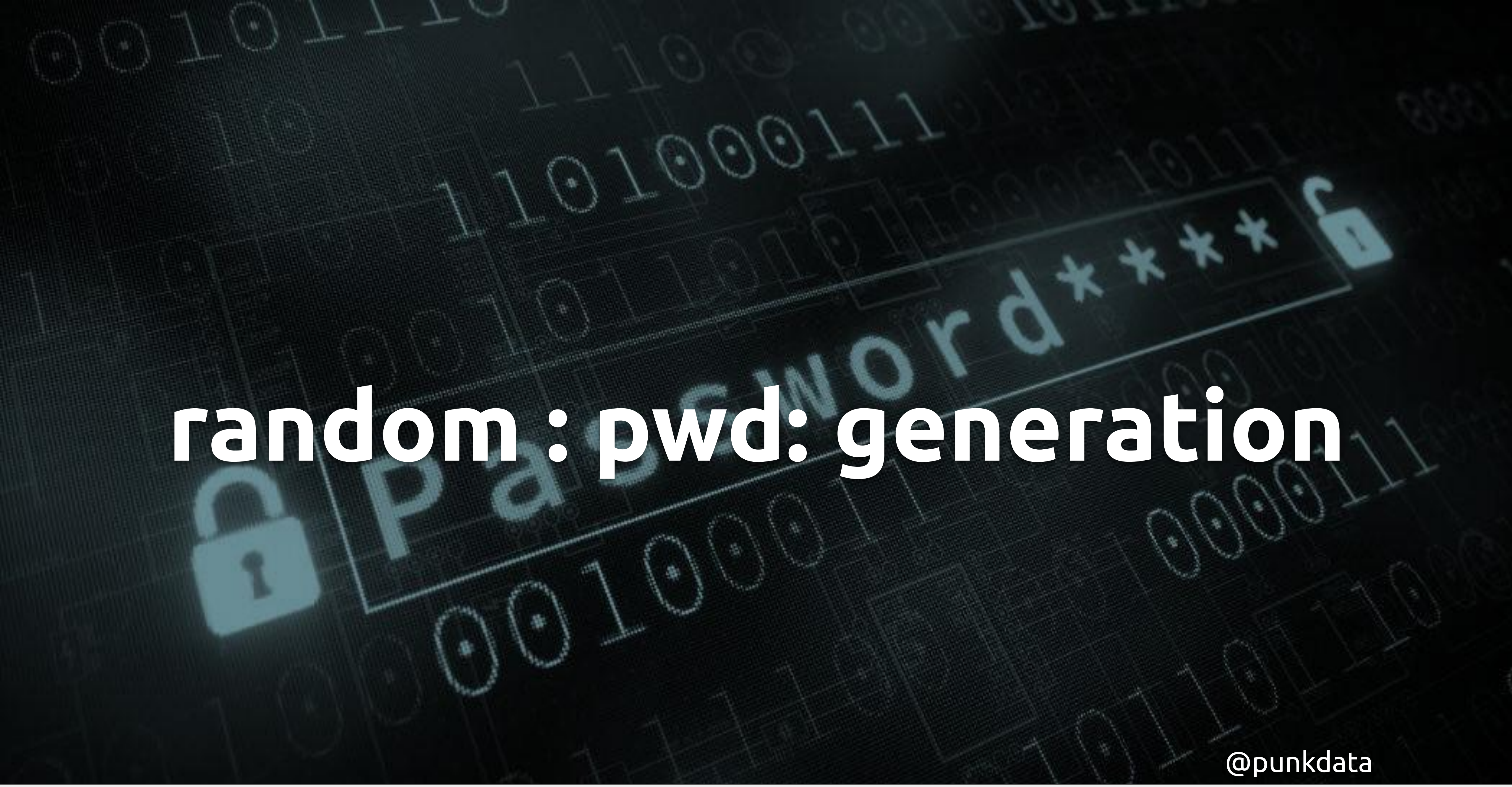
secrets management

@punkdata

tooling : protect : secrets



@punkdata



random : pwd: generation

@punkdata

auto : rotate : passwords

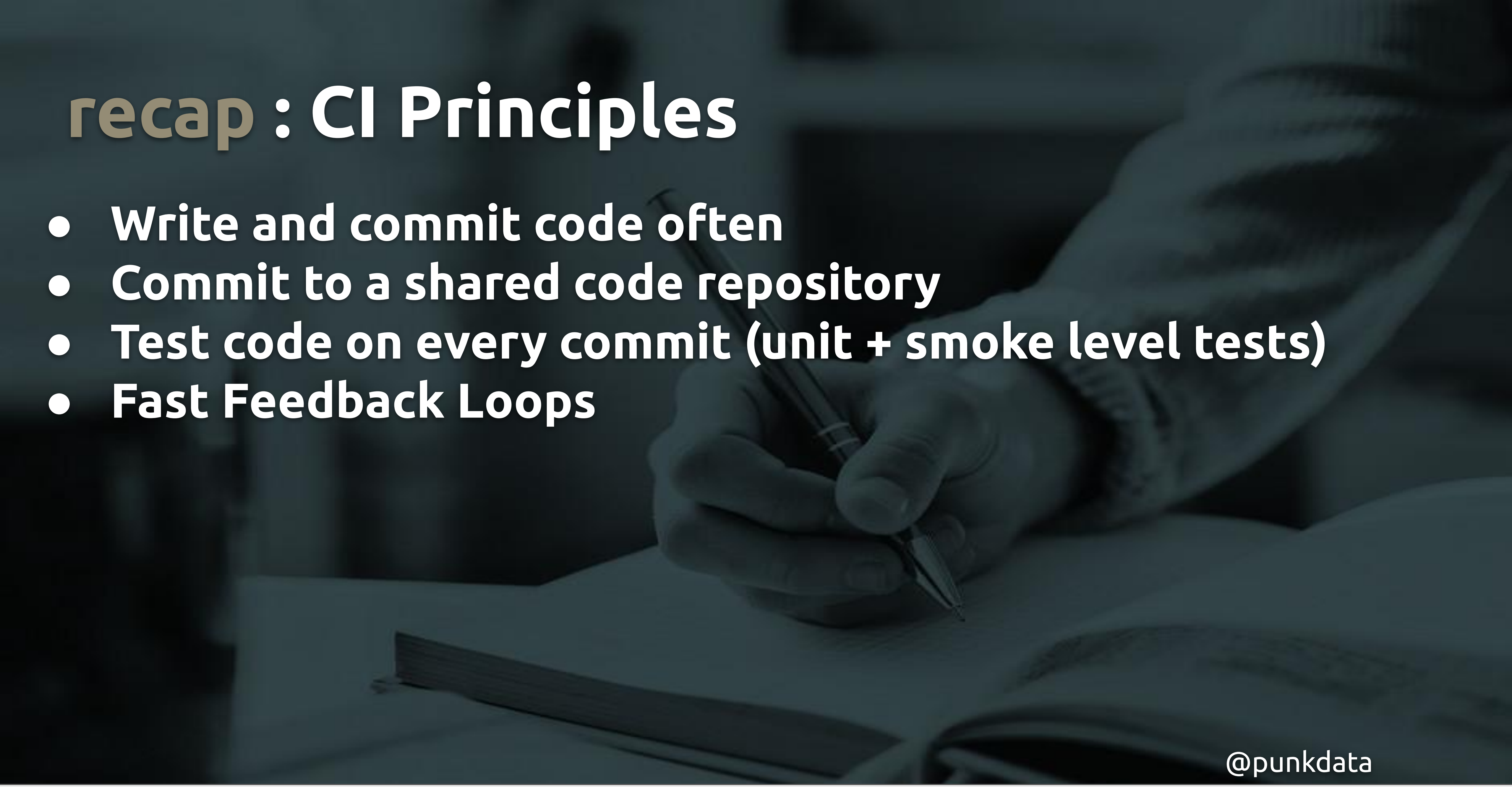
@punkdata

granular : access : controls

@punkdata

recap

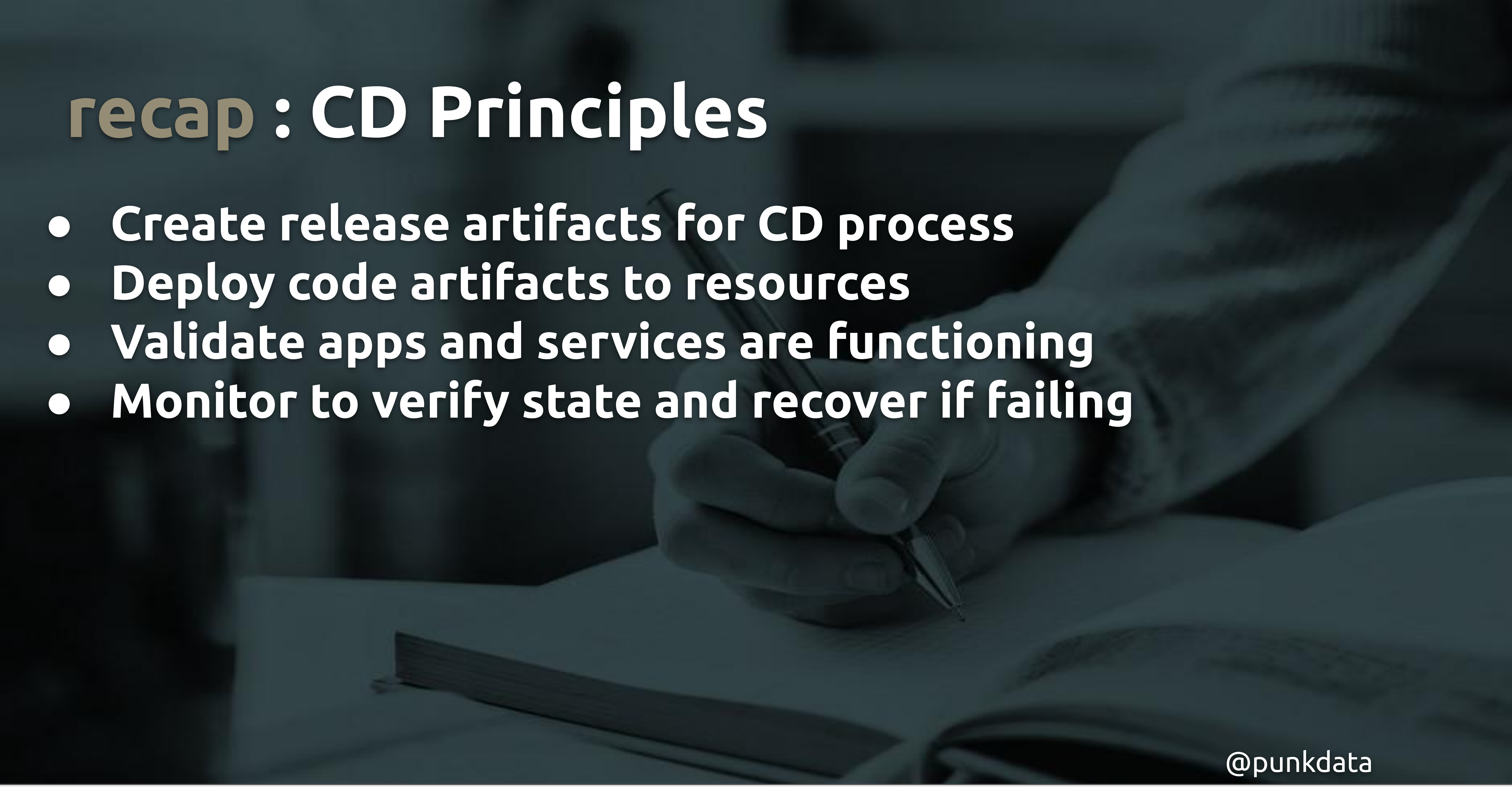
@punkdata



recap : CI Principles

- Write and commit code often
- Commit to a shared code repository
- Test code on every commit (unit + smoke level tests)
- Fast Feedback Loops

@punkdata



recap : CD Principles

- Create release artifacts for CD process
- Deploy code artifacts to resources
- Validate apps and services are functioning
- Monitor to verify state and recover if failing

@punkdata

recap : DevSecOps

- Shift Security Left
- App Scanning within Pipelines
- Container Scanning within Pipelines

@punkdata

recap : Securing : Pipelines

- implement secrets management policies and tooling
- integrate secrets management tools into pipelines
- generate strong random secrets + credentials
- auto rotate + replace secrets used in pipelines

@punkdata

thank : you

Angel Rivera

Developer Advocate

circleci

@punkdata