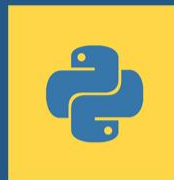




Ruslan Korniichuk

**Architecture as Code (AaC)
with Python**



**Pytech
Summit**
All about Python **2022**

Sala C

Godz. 10:00

allegro



Unit8™

Luxoft
A DXC Technology Company





Architecture as Code (AaC) with Python

Presented by: Ruslan Kornichuk 08.12.2022

A woman with blonde hair, wearing a dark jacket and a red and blue patterned scarf, is smiling and looking down at a white smartphone in her hands. She is wearing large gold hoop earrings. The background is a blurred outdoor setting with other people and buildings.

Introduction



Infrastructure as Code

Search term



Architecture as Code

Search term



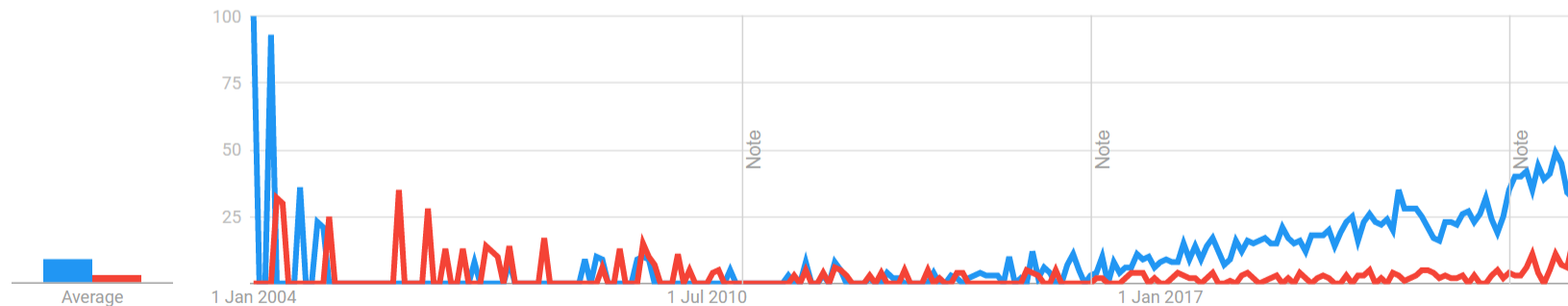
Add comparison

United States ▼

2004 – present ▼

All categories ▼

Web Search ▼

Interest over time 



● Infrastructure as ...

Search term

● Diagram as Code

Search term

● Architecture as Co...

Search term

+ Add comparison

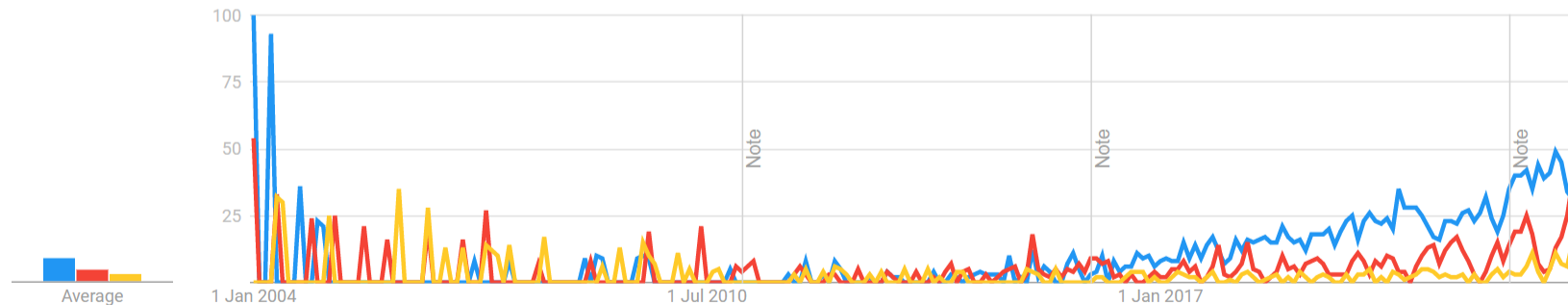
United States ▼

2004 – present ▼

All categories ▼

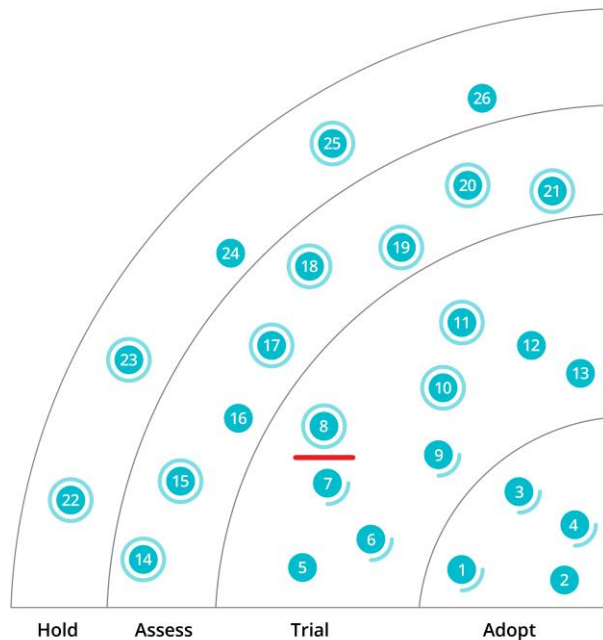
Web Search ▼

Interest over time ?



Technology Radar by ThoughtWorks

October 2020



Adopt

1. Dependency drift fitness function
2. Run cost as architecture fitness function
3. Security policy as code
4. Tailored service templates

Trial

5. Continuous delivery for machine learning (CD4ML)
6. Data mesh
7. Declarative data pipeline definition
8. Diagrams as code
9. Distroless Docker Images
10. Event interception
11. Parallel run with reconciliation
12. Use "remote native" processes and approaches
13. Zero trust architecture

Assess

14. Bounded low-code platforms
15. Browser-tailored polyfills
16. Decentralized identity
17. Kube-managed cloud services
18. Open Application Model (OAM)
19. Secure enclaves
20. Switchback experimentation
21. Verifiable credentials

Hold

22. Apollo Federation
23. ESBs in API Gateway's clothing
24. Log aggregation for business analytics
25. Micro frontend anarchy
26. Productionizing notebooks

Source: [thght.works/3frtPKE](https://thoughtworks.com/3frtPKE)

Infrastructure as Code vs Diagram as Code vs Architecture as Code

Infrastructure as Code (IaC) is the managing and provisioning of infrastructure through code instead of through manual processes.

Tools: Terraform, CloudFormation, Serverless Application Model (SAM), Cloud Development Kit (CDK).

Infrastructure as Code vs Diagram as Code vs Architecture as Code

Infrastructure as Code (IaC) is the managing and provisioning of infrastructure through code instead of through manual processes.

Tools: Terraform, CloudFormation, Serverless Application Model (SAM), Cloud Development Kit (CDK).

Diagram as Code is the creation of diagrams using code, and without any design tools.

Tools: Diagrams, Ilograph, Mermaid, Structurizr DSL.

Infrastructure as Code vs Diagram as Code vs Architecture as Code

Infrastructure as Code (IaC) is the managing and provisioning of infrastructure through code instead of through manual processes.

Tools: Terraform, CloudFormation, Serverless Application Model (SAM), Cloud Development Kit (CDK).

Diagram as Code is the creation of diagrams using code, and without any design tools.

Tools: Diagrams, Ilograph, Mermaid, Structurizr DSL.

Architecture as Code (AaC) is the prototyping and visualization of system architecture using code.

Tools: Diagrams (Python).

Diagram as Code tools

- AsciiDoctor Diagram: set of AsciiDoctor extensions
- CloudGram: Domain-Specific Language (DSL)
- D2: DSL
- Graphviz: DOT language
- Ilograph: YAML language
- Mermaid: Mermaid's syntax (Markdown-inspired)
- PlantUML: PlantUML language
- Structurizr DSL: DSL
- WebSequenceDiagrams: DSL

Static vs interactive

Static diagrams (e.g., PNG, SVG):

- D2
- Graphviz
- Mermaid
- PlantUML
- WebSequenceDiagrams

Interactive diagrams (e.g., browser-based):

- CloudGram
- Ilograph
- Structurizr DSL

```

@startuml
nwdiag {
  internet [ shape = cloud];
  internet -- router;

  network proxy {
    router;
    app;
  }
  network default {
    app;
    db;
  }
  group {
    color = "pink";
    app;
    db;
  }
}
@enduml

```

//www.plantuml.com/plantuml/png/LSrH3e8m30RwPtUAXdTEG4YuX_6Xiec4iJpb3Hh3tPtArFNjN_VzxPQ84dNs9gnsn04U1jAC8Je9B18HbYkoinPwJsfFJRckQn3I51hpNgItbMG25hhTNrtxv4yv8_CdR0Mpxe8gumuFoFmZz1m1XjJ8tmCzUH9eeU8nJ5KschTuCvqBLcV_1000

Decode URL

Submit

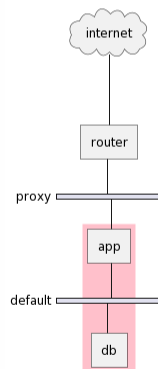


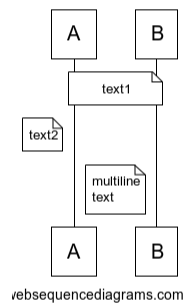
PNG SVG ASCII Art

online diagrams 175,297,041

current rate 432 diag. per minute

peak rate 952 diag. per minute





```
1 flowchart LR
2   A[Hard] -->|Text| B(Round)
3   B --> C(Decision)
4   C -->|One| D[Result 1]
5   C -->|Two| E[Result 2]
```

> Sample Diagrams

> History

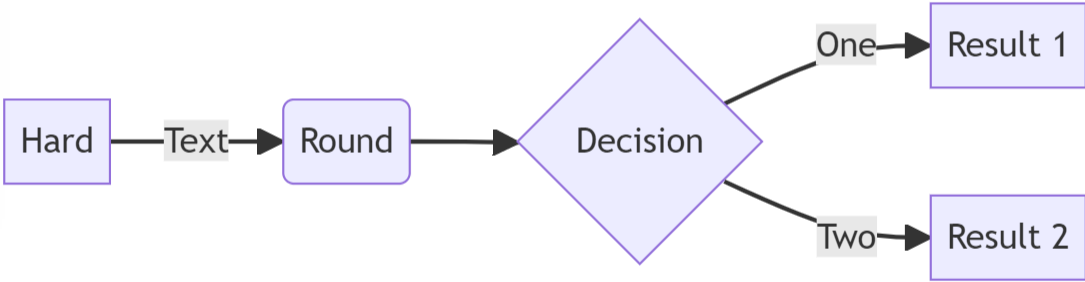
▼ Actions

COPY IMAGE TO CLIPBOARD

PNG SVG PNG SVG KROKI PNG size ☒ Auto ☐ Width ☐ Height

COPY MARKDOWN

LOAD GIST



The Structurizr DSL (as mentioned on the ThoughtWorks Tech Radar - Techniques - Diagrams as code) allows you to create multiple diagrams based on the C4 model, in multiple output formats, from a single DSL source file. **Some features (ldocs , ladsrs , lscript , etc) are unavailable on this demo page - see Help - DSL for details.**

DSL language reference

Upload

</>

⌕

Render

```
1 workspace {
2
3   model {
4     u = person "User"
5     s = softwareSystem "Software System" {
6       webapp = container "Web Application" "" "Spring Boot"
7       database = container "Database" "" "Relational database schema"
8     }
9
10    u -> webapp "Uses"
11    webapp -> database "Reads from and writes to"
12
13    live = deploymentEnvironment "Live" {
14      deploymentNode "Amazon Web Services" {
15        tags "Amazon Web Services - Cloud"
16
17        deploymentNode "US-East-1" {
18          tags "Amazon Web Services - Region"
19
20          route53 = infrastructureNode "Route 53" {
21            tags "Amazon Web Services - Route 53"
22          }
23
24          elb = infrastructureNode "Elastic Load Balancer" {
25            tags "Amazon Web Services - Elastic Load Balancing"
26          }
27
28          deploymentNode "Amazon EC2" {
29            tags "Amazon Web Services - EC2"
30
31            deploymentNode "Ubuntu Server" {
32              webApplicationInstance = containerInstance webapp
33            }
34          }
35
36          deploymentNode "Amazon RDS" {
37            tags "Amazon Web Services - RDS"
38
39            deploymentNode "MySQL" {
40              tags "Amazon Web Services - RDS MySQL instance"
41
42              containerInstance database
43            }
44          }
45        }
46
47        route53 -> elb "Forwards requests to" "HTTPS"
48        elb -> webApplicationInstance "Forwards requests to" "HTTPS"
49      }
50    }
51
52    views {
53      deployment s live {
54        include *
55        autoLayout lr
56      }
57
58      styles {
59        element "Element" {
60          shape RoundedBox
61          background #ffffff
62          color #000000
63        }
64      }
51
```

Structurizr DSL version: 1.21.0 | Structurizr CLI | Structurizr Lite | Themes | Language reference | Cookbook | Help

Examples: Getting started | Big Bank plc | Amazon Web Services

Structurizr
Diagram

Structurizr
Graph

Export
PlantUML

Export
C4-PlantUML

Export
Mermaid

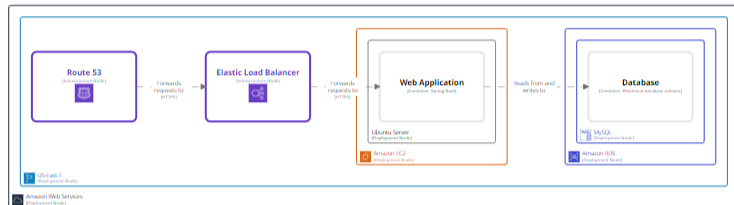
Export
DOT

Export
WebSequenceDiagrams

Export
Ilograph

[Deployment] Software System - Live (#SoftwareSystem-Live-Deployment)

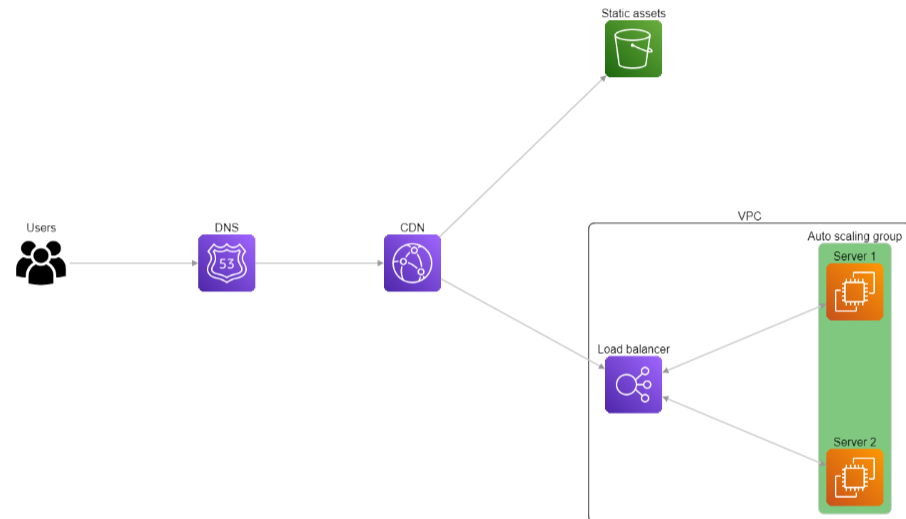
(diagram not editable; automatic layout enabled - help)

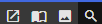


Structured Software Engineering - 1.2.1

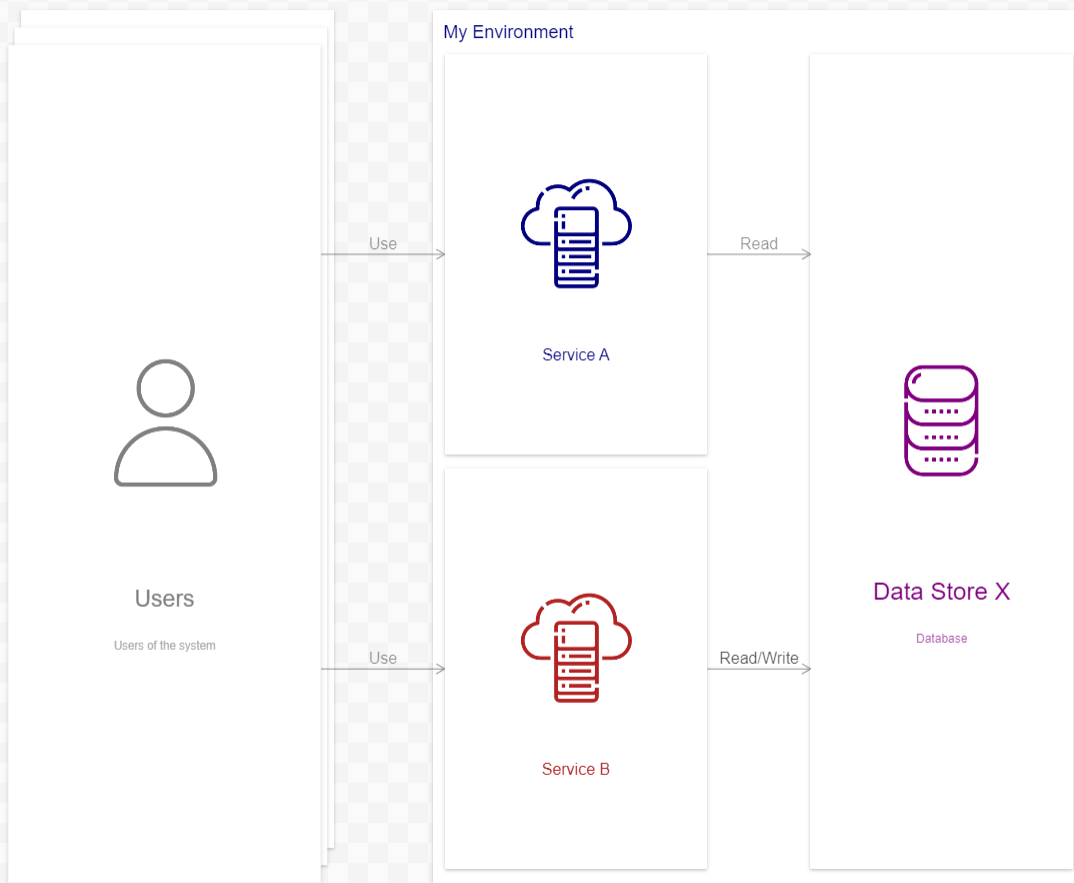


```
1 diagram "example" [direction=lr] {
2   // creating the nodes
3   generic.users Users;
4   aws.route53 DNS;
5   aws.cloudfront cf [label="CDN"];
6   aws.s3 s3 [label="Static assets"];
7
8   group vpc [label="VPC",style=solid,stroke=black,opacity=0] {
9     aws.elb load_balancer [label="Load balancer"];
10
11     group asg [label="Auto scaling group",style=solid,fill="#80c880"] {
12       aws.ec2 server1 [label="Server 1"];
13       aws.ec2 server2 [label="Server 2"];
14     }
15   }
16
17   // creating the edges
18   Users -> DNS -> cf -> load_balancer <=> asg;
19   cf -> s3;
20 }
```





```
1 resources:
2 - name: Users
3   subtitle: Users of the system
4   color: Gray
5   style: plural
6   icon: Networking/user.svg
7
8 - name: My Environment
9   subtitle: Environment
10  color: navy
11  children:
12 - name: Service A
13   subtitle: Service
14   icon: Networking/cloud-server.svg
15
16 - name: Service B
17   subtitle: Service
18   color: Firebrick
19   icon: Networking/cloud-server.svg
20
21 - name: Data Store X
22   subtitle: Database
23   color: Purple
24   description: This is a description
25   icon: Networking/database.svg
26
27 perspectives:
28 - name: Dependency
29   relations:
30 - from: Users
31   to: Service A, Service B
32   label: Use
33
34 - from: Service A
35   to: Data Store X
36   label: Read
37
38 - from: Service B
39   to: Data Store X
40   label: Read/Write
41   description: Labels can be given extended
42     descriptions, like this
43
44 notes: |-
45   Welcome to your Ilograph diagram.
46
47   Your diagram comes pre-populated with
48   resources and a sample perspective.
49
50 **Need Help**? Read [Creating your first
51   Ilograph diagram in 5 minutes](https://www.
52   .ilograph.com/docs/editing/tutorial/#making
53   -some-simple-changes). Additional resources
54   are available on the [Ilograph Docs](https
55   ://www.ilograph.com/docs/editing/).
```



Perspective notes

Welcome to your Ilograph diagram.

Your diagram comes pre-populated with resources and a sample perspective.

Need Help? Read [Creating your first Ilograph diagram in 5 minutes](#).

Additional resources are available on the [Ilograph Docs](#).

Example Diagrams

[AWS Diagram \(Relation\)](#)

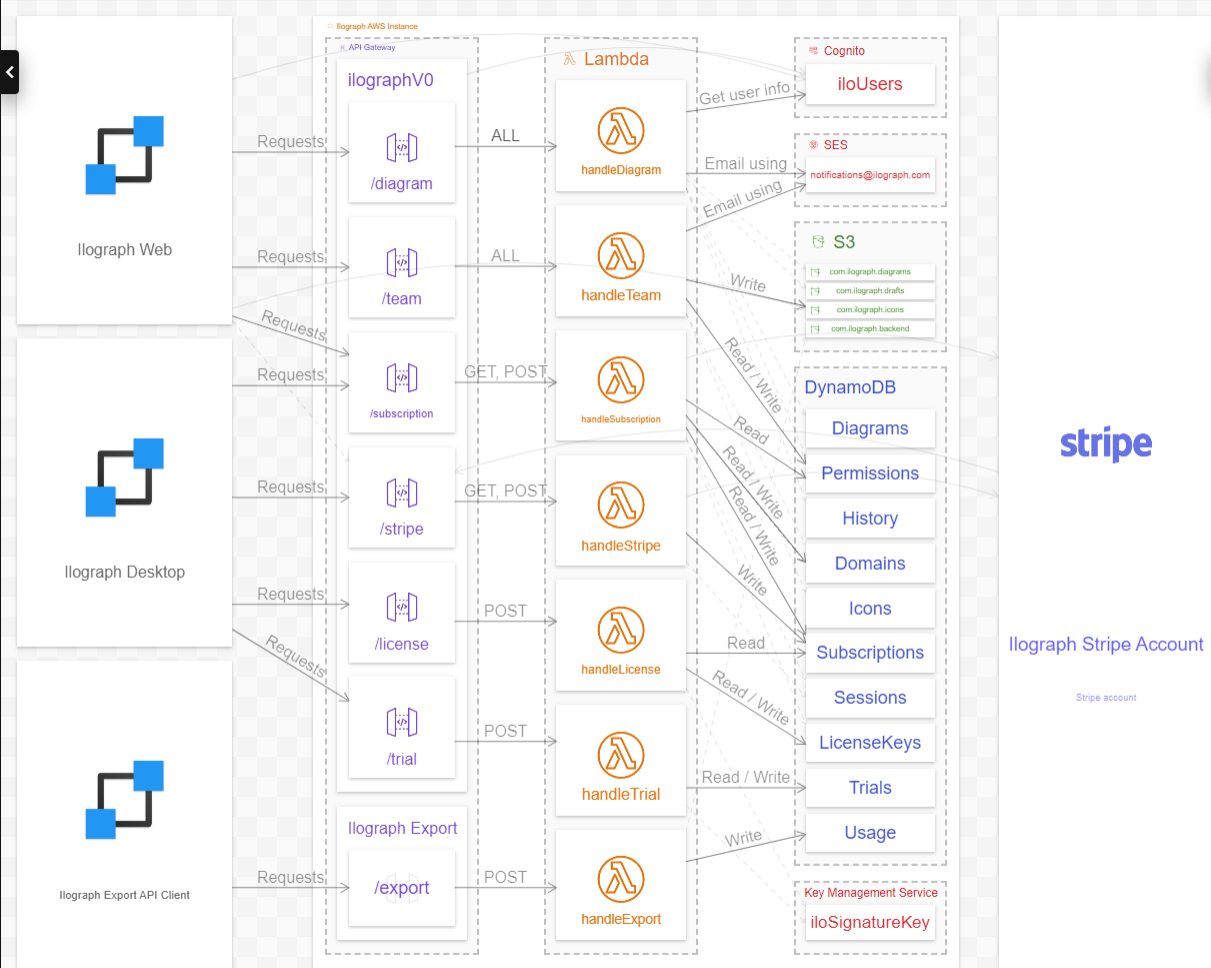
[AWS Diagram \(Sequence\)](#)

[Datacenter Diagram](#)



This diagram is read-only. [Click here](#) to create a new diagram or [log in](#) now.

```
1 imports:
2 - from: illograph/aws
3 namespace: AWS
4
5 resources:
6 - name: Ilograph Web
7   subtitle: Web Application
8   description: Advanced diagramming web application for creating interactive, multi-perspective diagrams in the browser
9   icon: Companies/illograph.png
10
11 - name: Ilograph Desktop
12   subtitle: Desktop Application
13   description: Advanced diagramming [desktop application](https://www.illograph.com/desktop/) for creating interactive, multi-perspective diagrams
14   icon: Companies/illograph.png
15
16 - name: Ilograph Export API Client
17   subtitle: NPM Package and CLI Tool
18   description: Package and CLI [tool](https://www.illograph.com/export-api/index.html) for exporting Ilograph diagrams to standalone HTML files
19   icon: Companies/illograph.png
20
21 - name: User
22   subtitle: Application User
23   description: End user of Ilograph
24   icon: AWS_General/User.svg
25
26 - name: Ilograph AWS Instance
27   instanceOf: AWS::AWS Instance
28   children:
29     - name: CloudFront
30       instanceOf: AWS::CloudFront
31       description: A CDN which speeds up the distribution of content to end users
32       children:
33         - name: d3dxxxxx.cloudfront.net
34           instanceOf: AWS::CloudFront::Distribution
35
36         - name: d2wxxxxx.cloudfront.net
37           instanceOf: AWS::CloudFront::Distribution
38
39         - name: d3exxxxx.cloudfront.net
40           instanceOf: AWS::CloudFront::Distribution
41
42 - name: API Gateway
43   instanceOf: AWS::ApiGateway
44   description: Service for creating REST APIs for Lambdas and other services
45   children:
46     - name: illographV0
47       instanceOf: AWS::ApiGateway::RestApi
48       description: The Ilograph REST API
49       children:
50         - id: diagram
51           name: /diagram
52           instanceOf: AWS::ApiGateway::Resource
53           description: Resource for Ilograph diagrams, their drafts, and their
```



Start Walkthrough

Perspective notes

This perspective shows the run-time dependencies between resources when fulfilling requests to the Ilograph back-end.

The back-end uses a classic AWS Serverless architecture, with an *API Gateway* API calling *Lambda* functions backed by *DynamoDB* tables and *S3* buckets.

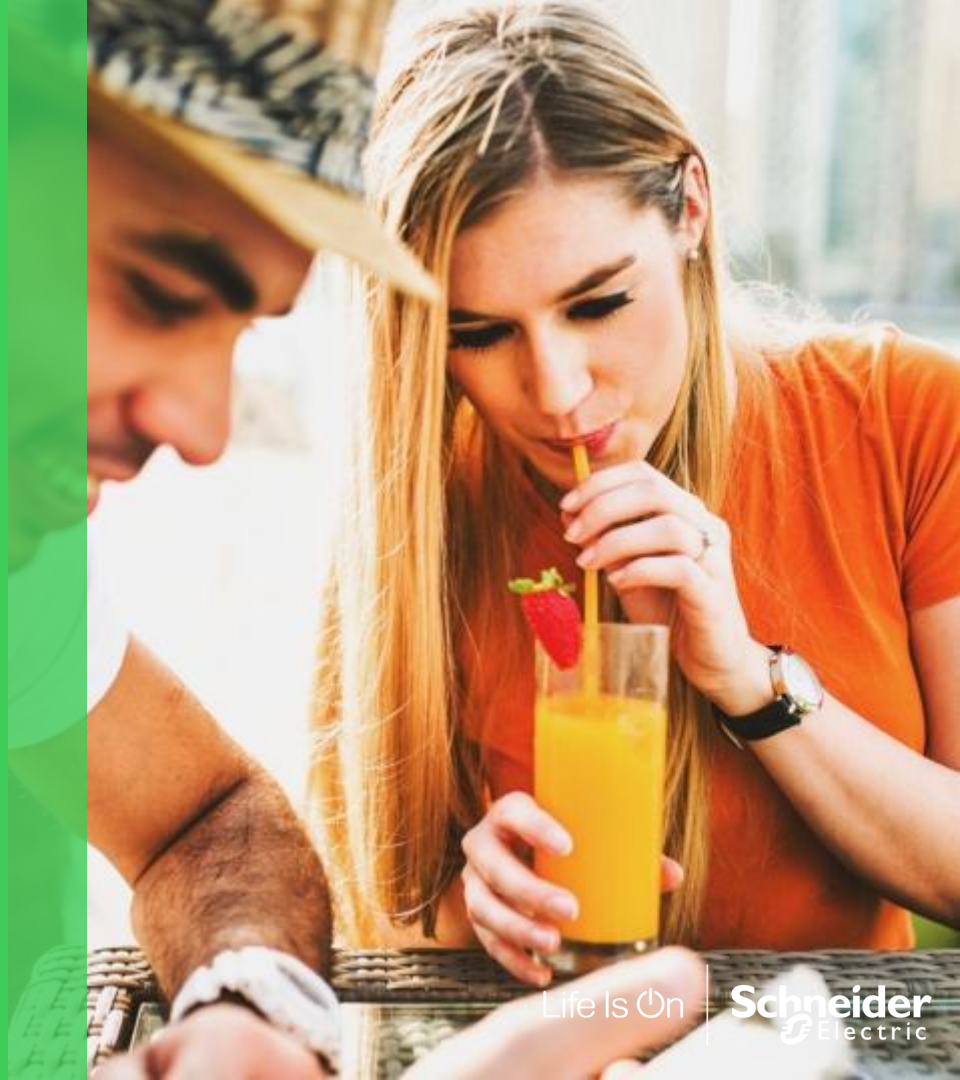
Requests related to purchasing and licenses also have an external dependency on Stripe.

RequestCodeDNS + OriginsGet diagramUpdate diagramInvite teammateUpload iconPurchase subscriptionClaim subscriptionActivate licer

"Terraform to CDK" trend

Migration from Terraform and CloudFormation to Cloud Development Kit (CDK).

JSON/YAML → Python



A male worker with a beard, wearing a yellow hard hat and safety glasses, is smiling broadly while holding a smartphone. He is wearing a dark blue work shirt. The background shows industrial equipment in a factory setting. A green horizontal bar is overlaid across the middle of the image, containing the text "Problem statement".

Problem statement

Do you use UML?

Do you use Microsoft Visio?

Do you use Draw.io?

Do you use whiteboard?

UML, PowerPoint, Visio, and even Draw.io are not good enough for engineers to create system architecture in a fun way.

A man and a woman are laughing and looking down at something out of frame in a room with a large window in the background. The woman is wearing a grey sweater and the man is wearing a dark hoodie. A green horizontal bar is overlaid on the image.

Solution

MAKE A DONATION TO OUR CHARITY FUND

The Charity Foundation "Come Back Alive" regularly and timely reports on its activities to all benefactors, state bodies and Ukrainian society.



READ THE REPORTS

Select the destination of your assistance

ARMY ASSISTANCE

SUPPORT THE FUND

All funds raised to help the army will be spent exclusively on purchases for the Armed Forces of Ukraine and other special forces in order to increase Ukraine's defense capabilities.

Payment method

PAYMENT BY CARD

TRANSFERS FROM ABROAD

CRYPTOCURRENCY

Regularity

SUBSCRIPTION

ONE TIME

Payment currency

UAH, ₴

USD, \$

EUR, €

Amount of donation*

€ 10

Email*

example@example.com

PAY

FAQ

HOW DO I KNOW WHAT'S REALLY GOING ON IN UKRAINE?



I DON'T WANT TO DONATE MONEY TO THE WAR. HOW CAN I HELP?



Architecture as Code (AaC) is easy to create, diff, version control, collaborate on, integrate into CI/CD.

A slide with no useful information at all

- Just filling in the gap between the last slide and the next one (which will be along in just a moment).
- No need to write this down, unless you feel compelled to do so.
- Nothing on this slide is examinable.
- In fact, I am not really sure why I bothered with it.

A slide with no useful information at all

- Just filling in the gap between the last slide and the next one (which will be along in just a moment).
- No need to write this down, unless you feel compelled to do so.
- Nothing on this slide is examinable.
- In fact, I am not really sure why I bothered with it.



Architecture as Code

pros

- track architecture diagram changes with version control
- visualize existing system architecture (e.g., Cloudiscovery tool)
- cloud agnostic (AWS, Azure, GCP, Alibaba, Oracle, etc.)
- reuse of source code and collaboration
- better for tax deduction
- expect coding 🚀

cons

- better for up-front design than long-lived documentation
- more for engineers
- whiteboard is better for prototyping

A woman with blonde hair, wearing a dark jacket and a red and blue patterned scarf, is smiling and looking down at a white smartphone in her hands. The background is a blurred indoor setting with other people.

Diagrams Python lib

master 10 branches 38 tags

Go to file

Add file

<> Code

 mingrammer chore: move black to dev-deps section	✓ 18a1976 11 days ago	🕒 489 commits
 .github	Setup dependabot. (#387)	2 years ago
 assets/img	fix: resize logo	3 years ago
 diagrams	feat(node): Add Portwork to onprem storage (#782)	29 days ago
 docker/dev	chore: fixing go get deprecation (#713)	4 months ago
 docs	feat(node): Add Portwork to onprem storage (#782)	29 days ago
 resources	feat: Update GCP icons (#666)	11 days ago
 scripts	docs: Add icons to docs (#499)	last month
 templates	docs: Add icons to docs (#499)	last month
 tests	feat: autolabel flag in diagrams that modifies label (#482)	last month
 website	feat: Update GCP icons (#666)	11 days ago
 .gitignore	Add programming languages and frameworks (#112)	3 years ago
 CHANGELOG.md	docs(changelog): update PR links	2 years ago
 CONTRIBUTING.md	docs: fix simple typo, cleaning -> cleaning (#724)	last month
 DEVELOPMENT.md	docs: fix typo (#730)	4 months ago
 LICENSE	license: create LICENSE.md	3 years ago
 README.md	docs: update README.md (#757)	last month
 autogen.sh	style: split the provider list by newline	last month
 config.py	feat(node): Add EMQX as an On-Prem Queue entry (#769)	29 days ago
 poetry.lock	chore: update poetry deps	last month
 pyproject.toml	chore: move black to dev-deps section	11 days ago

About

🔗 Diagram as Code for prototyping cloud system architectures

🔗 diagrams.mingrammer.com

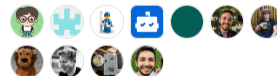
graphviz diagram architecture diagram-as-code

📖 Readme
📄 MIT license
⭐ 27.4k stars
👁 348 watching
🍴 1.7k forks

Releases 37

📦 v0.23.1 Latest
last month
+ 36 releases

Contributors 124


+ 113 contributors

Environments 1

🔗 github-pages Active

Languages

Diagrams Python lib

```
from diagrams import Diagram
from diagrams.aws.compute import EC2
from diagrams.aws.database import RDS
from diagrams.aws.network import ELB

with Diagram("Example 1", direction="TB", show=False):
    ELB("lb") >> [EC2("worker1"),
                   EC2("worker2"),
                   EC2("worker3"),
                   EC2("worker4"),
                   EC2("worker5")] >> RDS("events")
```

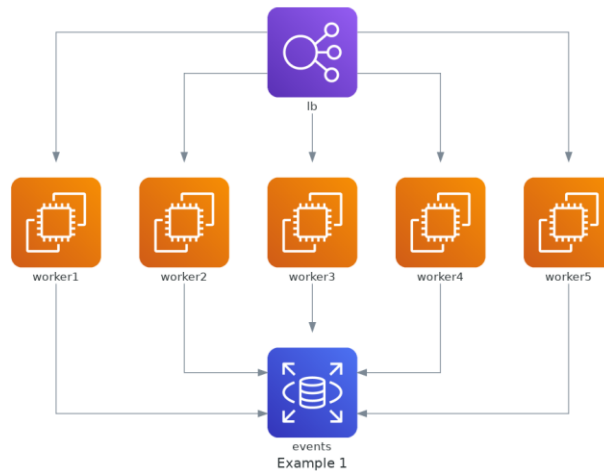


Diagram class

```
from diagrams import Diagram
from diagrams.aws.compute import EC2

graph_attr = {
    "bgcolor": "white"
}

with Diagram(
    name="Example 2",
    filename="example2",
    outformat=["png", "dot"],
    direction="LR",
    graph_attr=graph_attr,
    show=False):
    EC2("web")
```



web
Example 2

Nodes

```
from diagrams import Diagram
from diagrams.aws.compute import EC2

with Diagram("Example 3", filename="example3", show=False):
    EC2("web")
```



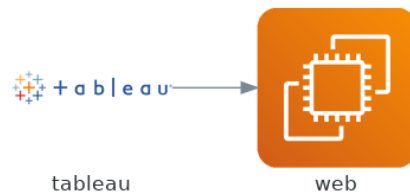
web
Example 3

Custom nodes

```
from diagrams import Diagram
from diagrams.aws.compute import EC2
from diagrams.custom import Custom

with Diagram("Example 4", filename="example4", show=False):
    tableau = Custom("tableau", "tableau_logo.png")

    tableau >> EC2("web")
```



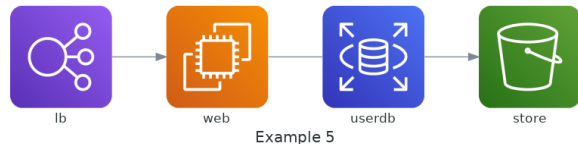
Example 4

Data flow

```
from diagrams import Diagram
from diagrams.aws.compute import EC2
from diagrams.aws.database import RDS
from diagrams.aws.network import ELB
from diagrams.aws.storage import S3

with Diagram("Example 5", filename="example5", show=False):
    elb = ELB("lb")
    ec2 = EC2("web")
    rds = RDS("userdb")
    s3 = S3("store")

    (elb >> ec2) - rds >> s3
```



Example 5

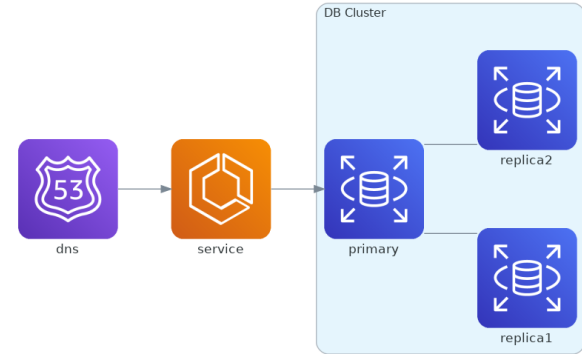
Clusters

```
from diagrams import Cluster, Diagram
from diagrams.aws.compute import ECS
from diagrams.aws.database import RDS
from diagrams.aws.network import Route53

with Diagram("Example 6", filename="example6", show=False):
    dns = Route53("dns")
    web = ECS("service")

    with Cluster("DB Cluster"):
        db_primary = RDS("primary")
        db_primary - [RDS("replica1"),
                     RDS("replica2")]

    dns >> web >> db_primary
```



Example 6

Nested clusters

```
from diagrams import Cluster, Diagram
from diagrams.aws.compute import ECS, EKS, Lambda
from diagrams.aws.database import Redshift
from diagrams.aws.integration import SQS
from diagrams.aws.storage import S3
```

```
with Diagram("Example 7", filename="example7", show=False):
    source = EKS("k8s source")
```

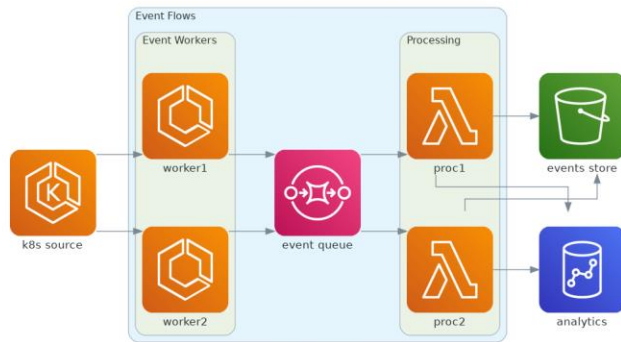
```
    with Cluster("Event Flows"):
        with Cluster("Event Workers"):
            workers = [ECS("worker1"),
                      ECS("worker2")]
```

```
    queue = SQS("event queue")
```

```
    with Cluster("Processing"):
        handlers = [Lambda("proc1"),
                   Lambda("proc2")]
```

```
    store = S3("events store")
    dw = Redshift("analytics")
```

```
    source >> workers >> queue >> handlers
    handlers >> store
    handlers >> dw
```



develop 4 branches 23 tags

Go to file

Add file

<> Code

leandrodamascena Merge pull request #175 from Cloud-Architects/python_39 ✓ fad132e on Jan 4, 2021 510 commits

.circleci	fixed deployment pipeline	2 years ago
bin	renamed tool, 2.1.0 release	3 years ago
clouddiscovery	v2.4.4	2 years ago
docs	Moved out AWS specific code to separate files	2 years ago
.codacy.yml	renamed tool, 2.1.0 release	3 years ago
.coveragerc	excluded tests from code coverage	3 years ago
.gitignore	restored readme assets	3 years ago
.pre-commit-config.yaml	added prospector to precommit, resolved all issues	3 years ago
.prospector.yaml	#97 added public and private subnet VPC case	2 years ago
.pylintrc	Added check if service is available in region #98	3 years ago
Dockerfile	#97 implemented generation of empty draw.io file	2 years ago
LICENSE	README issues and version bump	3 years ago
README.md	Version bump and readme adjusts	2 years ago
dependabot.yml	Added dependabot	3 years ago
pytest.ini	renamed tool, 2.1.0 release	3 years ago
requirements-dev.txt	trying making automated release - 2.1.0; added twine	3 years ago
requirements.txt	#127 code review	2 years ago
setup.cfg	#127 code review	2 years ago
setup.py	Version bump, python version updated and readme issues	2 years ago

README.md

About

The tool to help you discover resources in the cloud environment

python, iot, aws, aws-cli, vpc, diagrams, aws-iot, devops-tools, aws-monitoring, iam-policy, aws-services

Readme, Apache-2.0 license, 585 stars, 20 watching, 65 forks

Releases 19

cloud-discovery 2.4.4 Latest on Jan 4, 2021

+ 18 releases

Contributors 2

meshuga Patryk Orwat, leandrodamascena Leandro Damascena

Languages

Python 97.9%, HTML 1.6%, Other 0.5%

A man with a beard and mustache, wearing a yellow hard hat and clear safety glasses, is smiling broadly while looking at a smartphone he is holding with both hands. He is wearing a dark blue work shirt. The background is a blurred industrial setting with various pieces of machinery and equipment.

Closing slides

A smiling man with glasses on his head, wearing a pink shirt, is sitting at a desk in an office. He is looking towards the left. In the background, there are office shelves with various items, including a blue folder and a pen holder. The image is slightly blurred, giving it a professional yet approachable feel.

url24.me/PTS

main 1 branch 0 tags

Go to file Add file <> Code

 korniichuk Improve readability	4d14417 on Nov 3	🕒 4 commits
 .gitignore	Initial commit	last month
 LICENSE	Initial commit	last month
 README.md	Initial commit	last month
 d1_example.py	Improve readability	last month
 d2_diagram_class.py	Fix diagram 2	last month
 d3_nodes.py	Initial commit	last month
 d4_custom_nodes.py	Initial commit	last month
 d5_data_flow.py	Initial commit	last month
 d6_clusters.py	Initial commit	last month
 d7_nested_clusters.py	Initial commit	last month
 example1.png	Initial commit	last month
 example2.dot	Fix diagram 2	last month
 example2.png	Initial commit	last month
 example3.png	Initial commit	last month
 example4.png	Initial commit	last month
 example5.png	Initial commit	last month
 example6.png	Initial commit	last month
 example7.png	Initial commit	last month
 tableau_logo.png	Initial commit	last month

README.md

About

Architecture as Code (AaC) with Python or way to become your own boss

- 📖 Readme
- 📄 Unlicense license
- ☆ 0 stars
- 👁 2 watching
- 🍴 0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

Python 100.0%

[Home](#)[My Network](#)[Jobs](#)[Messaging](#)[Notifications](#)[Me](#)[Work](#)[Try Premium for free](#)

BE BRAVE LIKE UKRAINE



Ruslan Korniiichuk

Software Engineer (We're Hiring!)

Katowice, Śląskie, Poland · [Contact info](#)

500+ connections

[Open to](#)[Add profile section](#)[More](#)

Schneider Electric



Suggested for you

Private to you

Analytics

Private to you

145 profile views

Discover who's viewed your profile.

1,248 post impressions

Check out who's engaging with your posts.

56 search appearances

See how often you appear in search results.

Resources

Private to you

Creator mode **Off**

Get discovered, showcase content on your profile, and get access to creator tools

My network

See and manage your connections and interests.

Show all 5 resources →

About

[Edit public profile & URL](#)[Add profile in another language](#)

People also viewed



Michal Dulemba · 1st

Machine learning engineer at Egnyte, podcaster at nieliniowy.pl

[Message](#)

Daniel Telarik · 2nd

Software Engineering Manager - Leading software development with entrepreneurs...

[Connect](#)

Piotr Mundala · 1st

Global IS Domain Architect at ABB

[Message](#)

Tomasz Gajos · 2nd

UNIX IT Specialist w IBM Client Innovation Center

[Connect](#)

Jakub Michalik · 2nd

HR Projects Coordinator

[+ Follow](#)

Show more ▾

People you may know



Anna P.

Debt Collector with German & English

[Connect](#)

Dmytro Hurenok

Студент(ка) в уч. заведении Technikum elektryczny nr 2

[Connect](#)

Messaging



Questions?

Life Is On

Schneider
Electric

Life Is On

Schneider
Electric