Architecture as Code (AaC) with Python

or way to become your own boss

Presented by: Ruslan Korniichuk 03.11.2022

Introduction

Google Trends | Compare

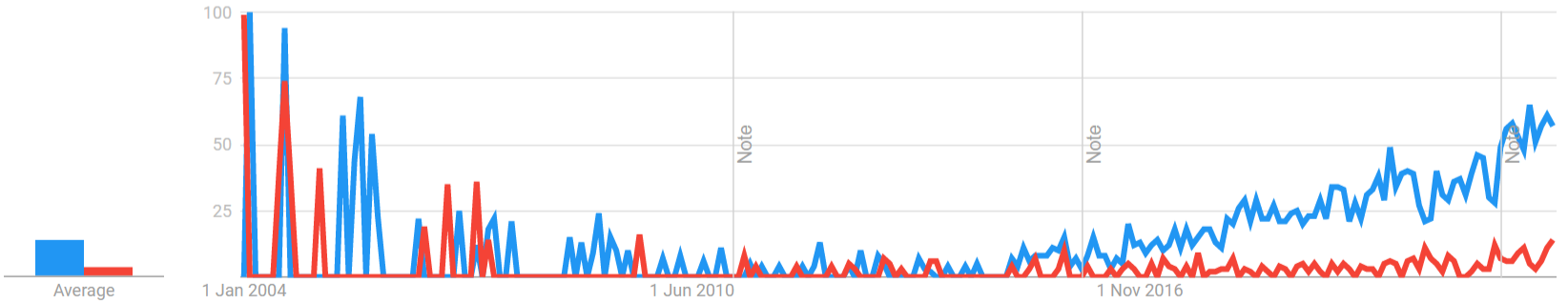
Infrastructure as Code
Search term

Architecture as Code
Search term

+ Add comparison

United States | 2004 – present | All categories | Web Search

Interest over time



Google Trends | Compare

Infrastructure as ...
Search term

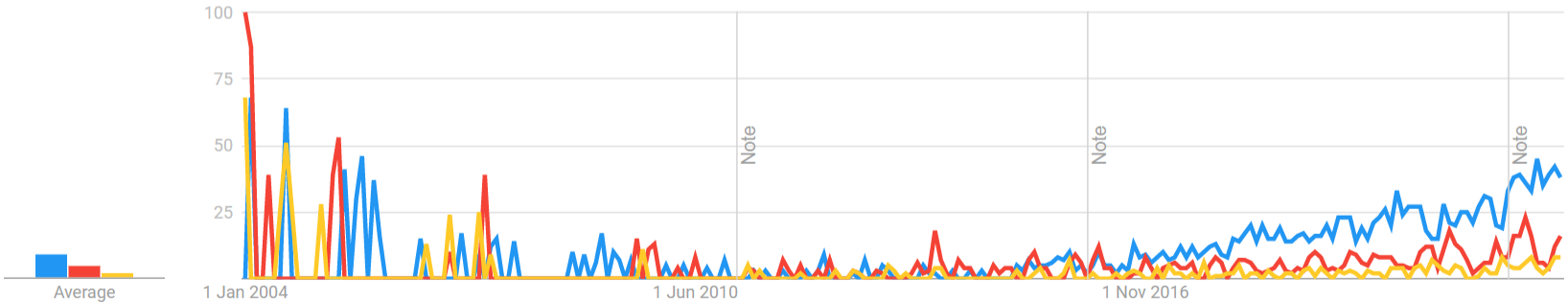
Diagram as Code
Search term

Architecture as Co...
Search term

+ Add comparison

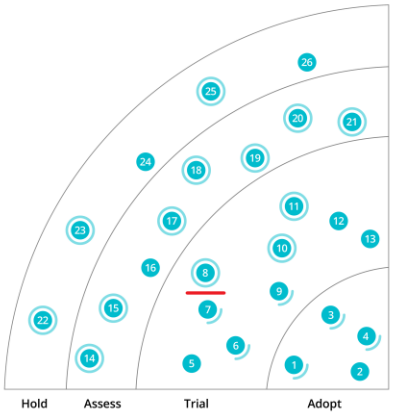
United States | 2004 – present | All categories | Web Search

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

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)
- 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):

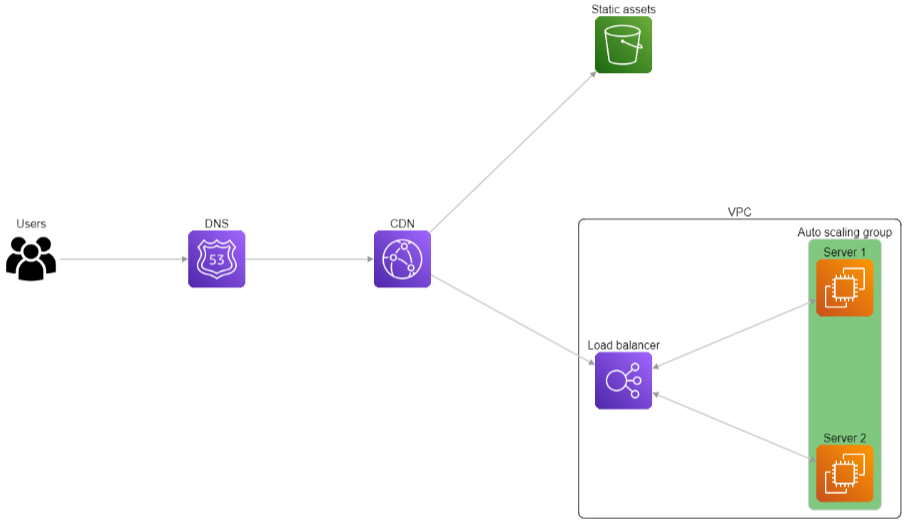
- Graphviz
- Mermaid
- PlantUML
- WebSequenceDiagrams

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

- CloudGram
- Ilograph
- Structurizr DSL

Embed Copy link png Save

```
diagram "example" [direction=lr] {
  // creating the nodes
  generic.users Users;
  aws.route53 DNS;
  aws.cloudfront cf [label="CDN"];
  aws.s3 s3 [label="Static assets"];

  group vpc [label="VPC",style=solid,stroke=black,opacity=0] {
      aws.elb load_balancer [label="Load balancer"];

      group asg [label="Auto scaling group",style=solid,fill="#80c880"] {
          aws.ec2 server1 [label="Server 1"];
          aws.ec2 server2 [label="Server 2"];
      }
  }

  // creating the edges
  Users -> DNS -> cf -> load_balancer <=> asg;
  cf -> s3;
}
```



Mermaid | Code | Config | Auto sync | DOCS

```
C4Context
title System Context diagram for Internet Banking System

Person(customerA, "Banking Customer A", "A customer of the bank, with personal bank accou
Person(customerB, "Banking Customer B")
Person_Ext(customerC, "Banking Customer C")
System(SystemAA, "Internet Banking System", "Allows customers to view information about th

Person(customerD, "Banking Customer D", "A customer of the bank, <br/> with personal bank

Enterprise_Boundary(b1, "BankBoundary") {

  SystemDb_Ext(SystemE, "Mainframe Banking System", "Stores all of the core banking inform

  System_Boundary(b2, "BankBoundary2") {
    System(SystemA, "Banking System A")
    System(SystemB, "Banking System B", "A system of the bank, with personal bank accounts

  }

  System_Ext(SystemC, "E-mail system", "The internal Microsoft Exchange e-mail system.")
  SystemDb(SystemD, "Banking System D Database", "A system of the bank, with personal bank

  Boundary(b3, "BankBoundary3", "boundary") {
    SystemQueue(SystemF, "Banking System F Queue", "A system of the bank, with personal ba
    SystemQueue_Ext(SystemG, "Banking System G Queue", "A system of the bank, with persona
  }
}

BiRel(customerA, SystemAA, "Uses")
```

Sample Diagrams

History

Actions

COPY IMAGE TO CLIPBOARD

PNG | SVG | PNG | SVG | KROKI | PNG size Auto Width Height

[![](https://mermaid.ink/img/pako:eNqtVVFP2zAQ_isnPzEpo2vLClTTpKYpaAS

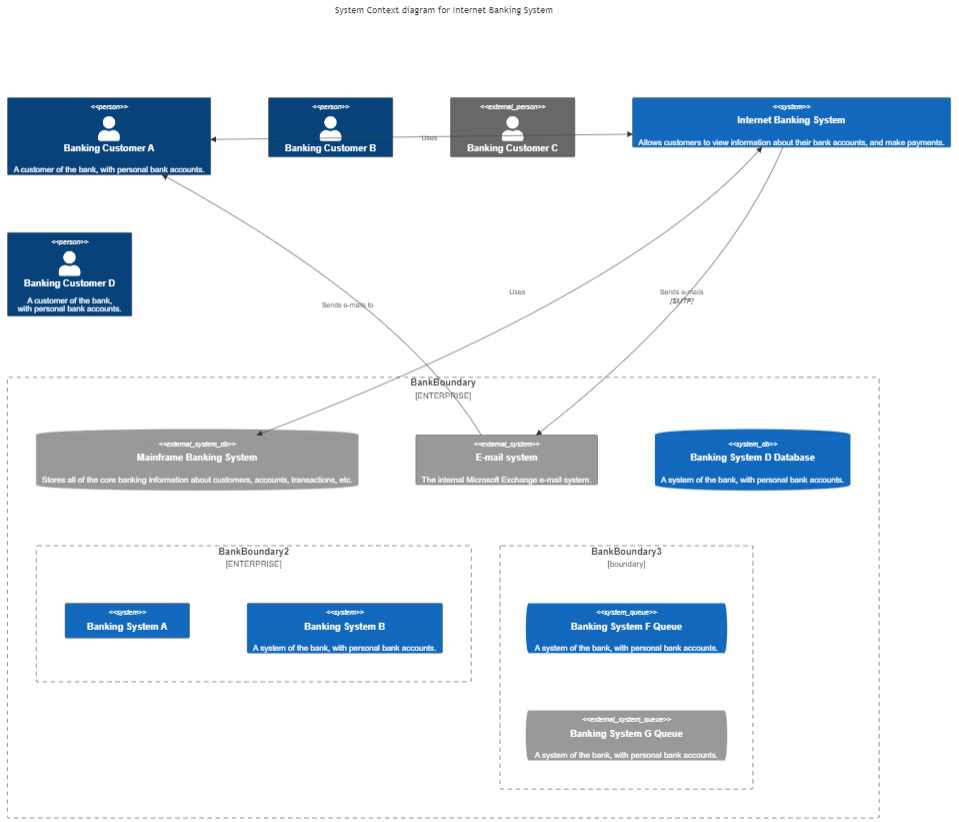
COPY MARKDOWN

https://gist.github.com/sidharthv96/6268a23e673a533dcb198f241fd7012a

LOAD GIST

This site is powered by Netlify

Diagram | Pan & Zoom | FULL SCREEN



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

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

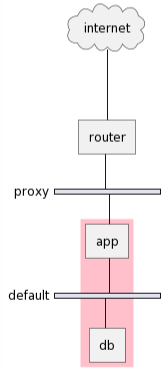
//www.plantuml.com/plantuml/png/LSrH3e8m30RWPtUAXdTEG4YuX_6Xiec4iJPb3Hh3tPtArFNjN_VzxPQ84dNs9gnsn04U1jAC8Je9Bi8HbYkoWnPwJsfFJRckQn3I51hpNgItbMG25hhTNrtxv4yv8_CdR0MpxeBgumuFoFmZzlmlXjJ8tmCzUH9eeU8nJ5KscHTuCvqBLcV_1000

Decode URL

Submit

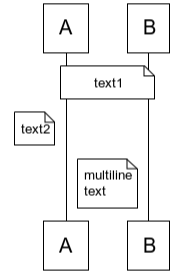
PNG SVG ASCII Art

online diagrams 175,297,041 current rate 432 diag. per minute peak rate 952 diag. per minute



File Edit View Style Help Upgrade Account

```
note over A,B: text1
note left of A: text2
note right of A
    multiline
    text
end note
```

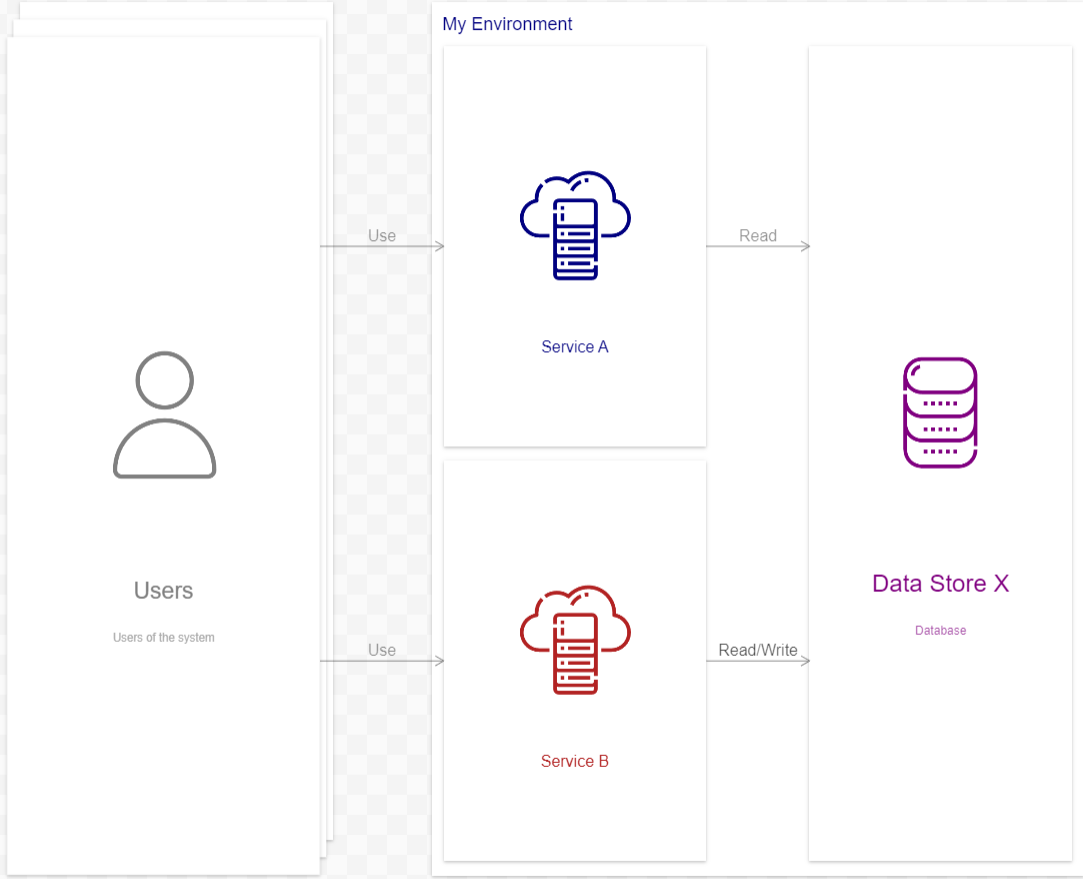


websequencediagrams.com

Untitled (unsaved)

Ln 8, Col 1 0

```
resources:
- name: Users
  subtitle: Users of the system
  color: Gray
  style: plural
  icon: Networking/user.svg

- name: My Environment
  subtitle: Environment
  color: navy
  children:
  - name: Service A
    subtitle: Service
    icon: Networking/cloud-server.svg

  - name: Service B
    subtitle: Service
    color: Firebrick
    icon: Networking/cloud-server.svg

  - name: Data Store X
    subtitle: Database
    color: Purple
    description: This is a description
    icon: Networking/database.svg

perspectives:
- name: Dependency
  relations:
  - from: Users
    to: Service A, Service B
    label: Use

  - from: Service A
    to: Data Store X
    label: Read

  - from: Service B
    to: Data Store X
    label: Read/Write
    description: Labels can be given extended
      descriptions, like this

  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](https://www
      .ilograph.com/docs/editing/tutorial/#making
      -some-simple-changes). Additional resources
      are available on the [Ilograph Docs](https
      ://www.ilograph.com/docs/editing/).

    **Example Diagrams**
    [AWS Diagram (Relation)](https://app.ilograph
      .com/demo.ilograph.Ilograph/Request)
    [AWS Diagram (Sequence)](https://app.ilograph
      .com/demo.ilograph.Ilograph/Get%2520diagram
      )
    [Datacenter Diagram](https://app.ilograph.com
      /demo.ilograph
      .Stack%2520Overflow%2520Architecture%2520\
      (2016\)/Traffic)
```



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

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

DSL language reference | Upload | </> | Render

```
workspace {

    model {
        u = person "User"
        s = softwareSystem "Software System" {
            webapp = container "Web Application" "" "Spring Boot"
            database = container "Database" "" "Relational database schema"
        }

        u -> webapp "Uses"
        webapp -> database "Reads from and writes to"

        live = deploymentEnvironment "Live" {
            deploymentNode "Amazon Web Services" {
                tags "Amazon Web Services - Cloud"

                deploymentNode "US-East-1" {
                    tags "Amazon Web Services - Region"

                    route53 = infrastructureNode "Route 53" {
                        tags "Amazon Web Services - Route 53"
                    }
                    elb = infrastructureNode "Elastic Load Balancer" {
                        tags "Amazon Web Services - Elastic Load Balancing"
                    }

                    deploymentNode "Amazon EC2" {
                        tags "Amazon Web Services - EC2"

                        deploymentNode "Ubuntu Server" {
                            webApplicationInstance = containerInstance webapp
                        }
                    }

                    deploymentNode "Amazon RDS" {
                        tags "Amazon Web Services - RDS"

                        deploymentNode "MySQL" {
                            tags "Amazon Web Services - RDS MySQL instance"

                            containerInstance database
                        }
                    }
                }
            }

            route53 -> elb "Forwards requests to" "HTTPS"
            elb -> webApplicationInstance "Forwards requests to" "HTTPS"
        }
    }

    views {
        deployment s live {
            include *
            autoLayout lr
        }

        styles {
            element "Element" {
                shape RoundedBox
                background #ffffff
                color #000000
            }
        }
```

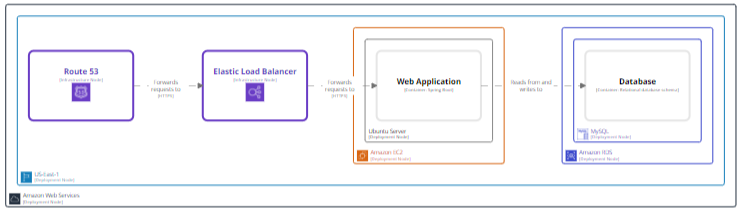
Structurizr DSL version: 1.21.0 | Structurizr CLI | Structurizr Lite | Themes | Language reference | Cookbook | Help

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)



"Terraform to CDK" trend

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

JSON/YAML → Python

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.

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

Payment method

PAYMENT BY CARD | TRANSFERS FROM ABROAD | CRYPTOCURRENCY

Regularity

SUBSCRIPTION | ONE TIME

Payment currency

UAH, ₴ | USD, $ | EUR, €

Amount of donation*

€ 10

Email*

name.surname@gmail.com

Buy with G Pay

or enter card details

Card number

XXXX XXXX XXXX XXXX

Expiry Date

MM/YY

CVV/CVC

•••

PAY

FAQ

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

Diagrams Python lib

mingrammer / diagrams

Public

Watch 352 | Fork 1.6k | Starred 26.3k

Code | Issues 237 | Pull requests 85 | Actions | Projects | Security | Insights

master | 4 branches | 37 tags | Go to file | Add file | Code

mingrammer bump: up to version 0.22.0 — 8348996 on 13 Sep — 451 commits

.github	Setup dependabot. (#387)	2 years ago
assets/img	fix: resize logo	3 years ago
diagrams	feat: Basic support for C4 model primitives. (#508)	2 months ago
docker/dev	chore: fixing go get deprecation (#713)	3 months ago
docs	feat: Basic support for C4 model primitives. (#508)	2 months ago
resources	feat(node): add GraphQL (#660)	3 months ago
scripts	feat(provider): added DigitalOcean provider (#621)	9 months ago
templates	docs: add more visibility to Custom node (#284) (#424)	2 years ago
tests	feat: Basic support for C4 model primitives. (#508)	2 months ago
website	feat: Basic support for C4 model primitives. (#508)	2 months ago
.gitignore	Add programming languages and frameworks (#112)	3 years ago
CHANGELOG.md	docs(changelog): update PR links	2 years ago
CONTRIBUTING.md	docs(contributing): add black tool requirement for autogen.sh (#434)	15 months ago
DEVELOPMENT.md	docs: fix typo (#730)	3 months ago
LICENSE	license: create LICENSE.md	3 years ago
README.md	feat: Basic support for C4 model primitives. (#508)	2 months ago
autogen.sh	feat(provider): added DigitalOcean provider (#621)	9 months ago
config.py	feat(node): add GraphQL (#660)	3 months ago
poetry.lock	chore(deps): bump graphviz from 0.17 to 0.19.1 (#635)	10 months ago
pyproject.toml	bump: up to version 0.22.0	2 months ago

About

Diagram as Code for prototyping cloud system architectures

diagrams.mingrammer.com

graphviz | diagram | architecture | diagram-as-code

Readme
MIT license
26.3k stars
352 watching
1.6k forks

Releases 36

v0.22.0 Latest
on 13 Sep

+ 35 releases

Contributors 108

+ 97 contributors

Environments 1

github-pages Active

Diagrams Python lib

```python
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")
```

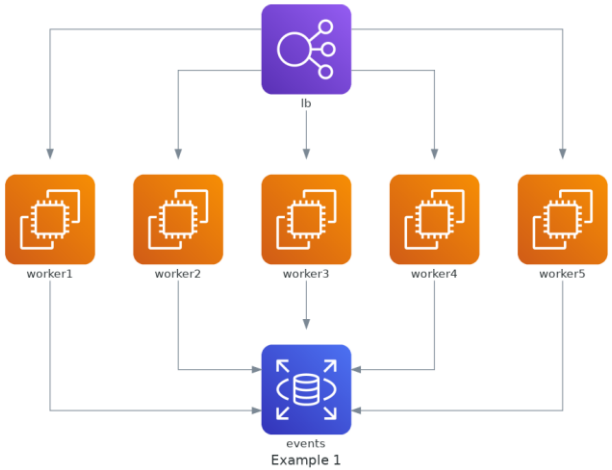


Diagram class

```
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")
```

Example 2

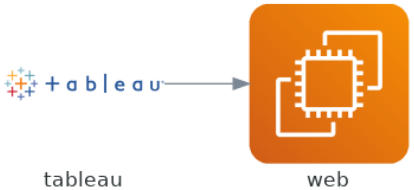
Nodes

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

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

Example 3

Custom nodes

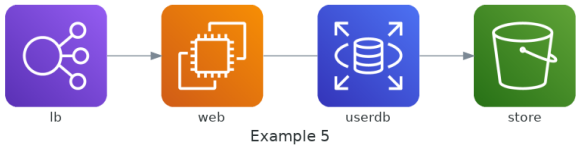
```
from diagrams import Diagram
from diagrams.aws.compute import EC2
from diagrams.custom import Custom

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

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



Example 4

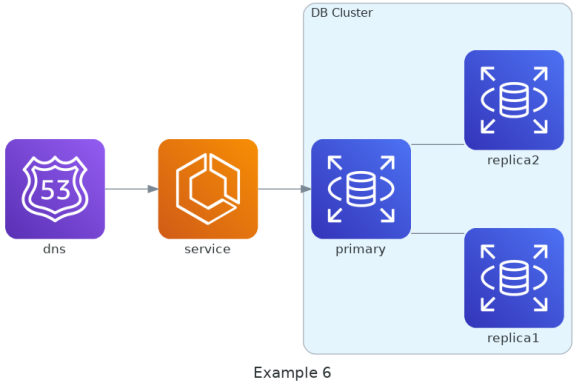
Data flow

```python
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
```



Clusters

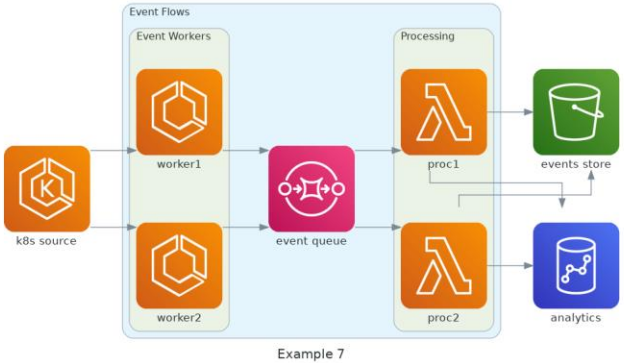
```
from diagrams import Cluster, Diagram
from diagrams.aws.compute import ECS
from diagrams.aws.database import RDS
from diagrams.aws.network import Route53

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
```



Example 7

Cloud-Architects / cloudiscovery

Public

Watch 20 | Fork 62 | Star 569

Code | Issues 23 | Pull requests 2 | Projects | Security | Insights

develop | 4 branches | 23 tags | Go to file | Add file | Code

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

.circleci	fixed deployment pipeline	2 years ago
bin	renamed tool, 2.1.0 release	2 years ago
cloudiscovery	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	2 years ago
.coveragerc	excluded tests from code coverage	2 years ago
.gitignore	restored readme assets	2 years ago
.pre-commit-config.yaml	added prospector to precommit, resolved all issues	2 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	2 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	2 years ago
pytest.ini	renamed tool, 2.1.0 release	2 years ago
requirements-dev.txt	trying making automated release - 2.1.0; added twine	2 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
569 stars
20 watching
62 forks

Releases 19

cloud-discovery 2.4.4 Latest
on 4 Jan 2021

+ 18 releases

Contributors 2

meshuga Patryk Orwat
leandrodamascena Leandro Damascena

Languages

Python 97.9% HTML 1.6% Other 0.5%

Questions?

Get AWS Certified

Watch an on-demand exam readiness class to receive 50% discount voucher for your exams.

AWS Certified Cloud Practitioner

AWS Certified Solutions Architect - Associate

AWS Certified Developer - Associate

aws

korniichuk / architecture-as-code-pycon-pl-2022 Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

korniichuk Initial commit 00350c1 1 minute ago 2 commits

.gitignore	Initial commit	41 minutes ago
LICENSE	Initial commit	41 minutes ago
README.md	Initial commit	41 minutes ago
d1_example.py	Initial commit	1 minute ago
d2_diagram_class.py	Initial commit	1 minute ago
d3_nodes.py	Initial commit	1 minute ago
d4_custom_nodes.py	Initial commit	1 minute ago
d5_data_flow.py	Initial commit	1 minute ago
d6_clusters.py	Initial commit	1 minute ago
d7_nested_clusters.py	Initial commit	1 minute ago
example1.png	Initial commit	1 minute ago
example2.dot	Initial commit	1 minute ago
example2.png	Initial commit	1 minute ago
example3.png	Initial commit	1 minute ago
example4.png	Initial commit	1 minute ago
example5.png	Initial commit	1 minute ago
example6.png	Initial commit	1 minute ago
example7.png	Initial commit	1 minute ago
tableau_logo.png	Initial commit	1 minute ago

README.md

About

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

Readme
Unlicense license
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Python 100.0%

Ruslan Korniichuk

Software Engineer (We're Hiring!)

Katowice, Śląskie, Poland · Contact info

500+ connections

Schneider Electric

Open to | Add profile section | More

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 entrepreneuri...
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