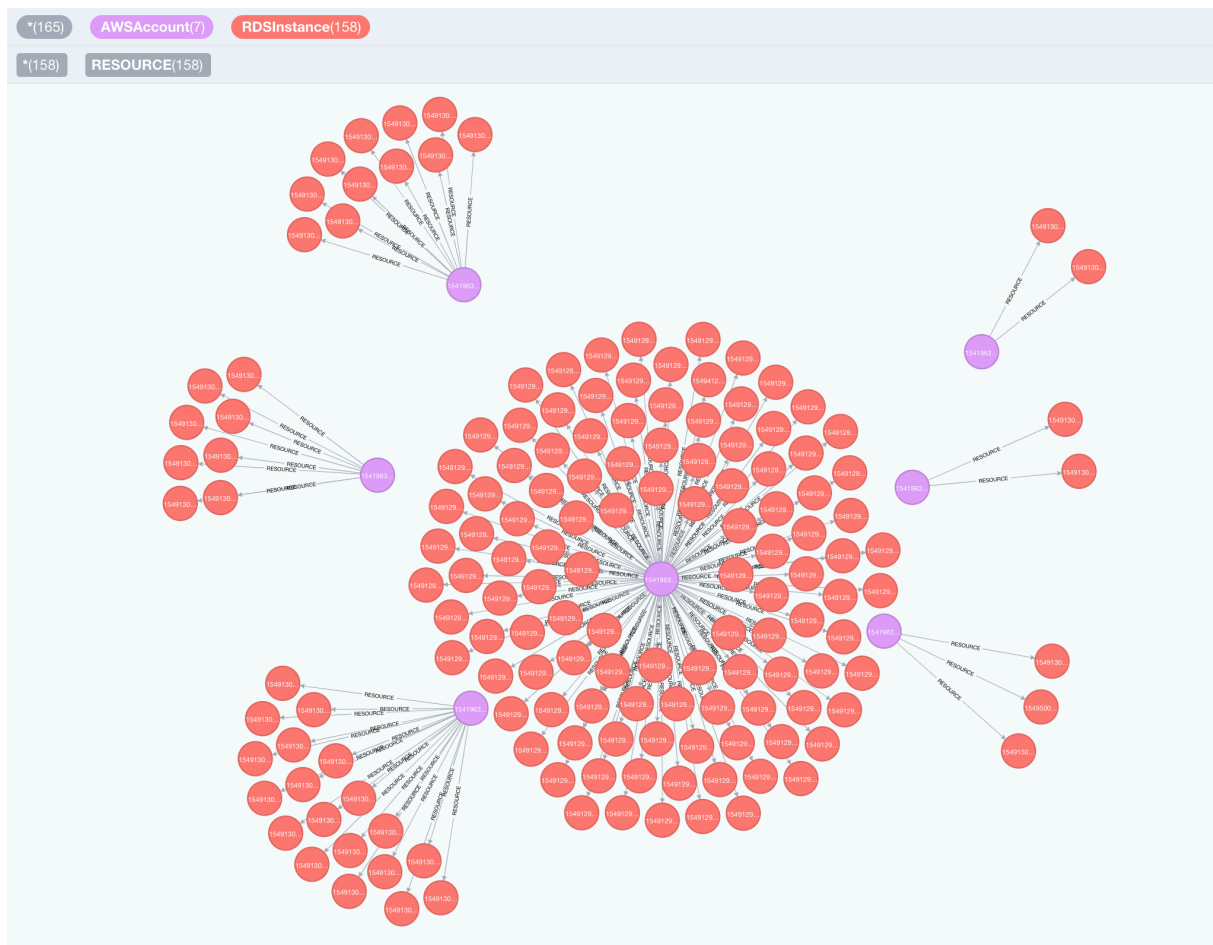




Cartography

Cartography is a Python tool that consolidates infrastructure assets and the relationships between them in an intuitive graph view powered by a Neo4j database.



Why Cartography?

Cartography aims to enable a broad set of exploration and automation scenarios. It is particularly good at exposing otherwise hidden dependency relationships between your service's assets so that you may validate assumptions about security risks.

Service owners can generate asset reports, Red Teamers can discover attack paths, and Blue Teamers can identify areas for security improvement. All can benefit from using the graph for manual exploration through a web frontend interface, or in an automated fashion by calling the APIs.

Cartography is not the only security graph tool out there, but it differentiates itself by being fully-featured yet generic and extensible enough to help make anyone better understand their risk exposure, regardless of what platforms they use. Rather than being focused on one core scenario or attack vector like the other linked tools, Cartography focuses on flexibility and exploration.

You can learn more about the story behind Cartography in our presentation at BSidesSF 2019.

Install and configure

Start here.

Supported platforms

- Amazon Web Services - API Gateway, Config, EC2, ECS, ECR, Elasticsearch, Elastic Kubernetes Service (EKS), DynamoDB, IAM, Inspector, KMS, Lambda, RDS, Redshift, Route53, S3, Secrets Manager, Security Hub, SQS, SSM, STS, Tags
- Google Cloud Platform - Cloud Resource Manager, Compute, DNS, Storage, Google Kubernetes Engine
- Google GSuite - users, groups
- Duo CRXcavator - Chrome extensions, GSuite users
- Oracle Cloud Infrastructure - IAM
- Okta - users, groups, organizations, roles, applications, factors, trusted origins, reply URIs
- Github - repos, branches, users, teams
- DigitalOcean
- Microsoft Azure - CosmosDB, SQL, Storage, Virtual Machine
- Kubernetes - Cluster, Namespace, Service, Pod, Container
- PagerDuty - Users, teams, services, schedules, escalation policies, integrations, vendors
- Crowdstrike Falcon - Hosts, Spotlight vulnerabilities, CVEs
- NIST CVE - Common Vulnerabilities and Exposures (CVE) data from NIST database
- Lastpass - users
- BigFix - Computers
- Duo - Users, Groups, Endpoints

Usage

Start with our tutorial. Our data schema is a helpful reference when you get stuck.

Community

- Join us on [#cartography](#) on the Lyft OSS Slack.
- Talk to us and see what we're working on at our monthly community meeting.
 - Meeting minutes are [here](#).
 - Recorded videos are [posted here](#).
- Our current project roadmap is [here](#).

Contributing

Thank you for considering contributing to Cartography!

Code of conduct

Legal stuff: This project is governed by Lyft's code of conduct. All contributors and participants agree to abide by its terms.

Bug reports and feature requests and discussions

Submit a GitHub issue to report a bug or request a new feature. If we decide that the issue needs more discussion - usually because the scope is too large or we need to make careful decision - we will convert the issue to a GitHub Discussion.

Developing Cartography

Get started with our developer documentation. Please feel free to submit your own PRs to update documentation if you've found a better way to explain something.

Sign the Contributor License Agreement (CLA) We require a CLA for code contributions, so before we can accept a pull request we need to have a signed CLA. Please visit our CLA service and follow the instructions to sign the CLA.

Who uses Cartography?

1. Lyft
2. Thought Machine
3. MessageBird
4. Cloudanix
5. ZeusCloud
6. {Your company here} :-)

If your organization uses Cartography, please file a PR and update this list. Say hi on Slack too!