CloudMapper

AWS environment analysis and visualization

Scott Piper
@0xdabbad00
SummitRoute.com
scott@summitroute.com

who_was_i

- 1st to publicly document reverse engineering of EMET
  - http://0xdabbad00.com/wp-content/uploads/2013/11/emet_4_1_uncovered.pdf
- Reverse engineered WER and built a product to detect failed exploits.
  - http://0xdabbad00.com/wp-content/uploads/2014/01/notes_on_wer.pdf
  - Based around the concept of John Lamberts MS08-067 story
- Built an EDR and app white-listing solution
  - Kernel driver + userland: https://github.com/SummitRoute/srepp_client

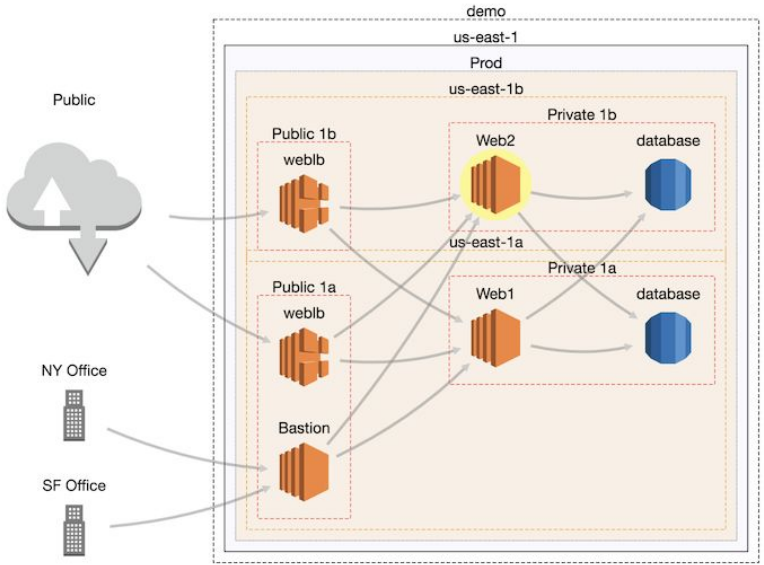
whoami

- Independent AWS security consultant

Scott Piper
@0xdabbad00
SummitRoute.com
scott@summitroute.com
Salt Lake City, Utah

- Creator of:
  - flAWS - http://flaws.cloud/
  - CloudMapper - https://github.com/duo-labs/cloudmapper
  - CloudTracker - https://github.com/duo-labs/cloudtracker

aws CERTIFIED Solutions Architect Associate
aws CERTIFIED Developer Associate
aws CERTIFIED SysOps Administrator Associate
aws CERTIFIED Solutions Architect Professional
aws CERTIFIED DevOps Engineer Professional
aws CERTIFIED Security Specialty



Web2

Summary

Type: ec2
ID: i-00000000000000002
Name: Web2

Details


```
{
    "Monitoring": {
        "State": "enabled"
    },
    "State": {
        "Code": 16,
        "Name": "running"
    },
    "EbsOptimized": true,
    "LaunchTime": "2017-12-01T00:00:00.000Z",
    "PrivateIpAddress": "10.0.3.1",
    "VpcId": "vpc-12345678",
    "StateTransitionReason": "",
    "InstanceId": "i-00000000000000002",
    "EnaSupport": true,
    "ImageId": "ami-00000001",
    "PrivateDnsName": "ip-10-0-3-1.ec2.internal",
    "KeyName": "web2",
```


Neighbors

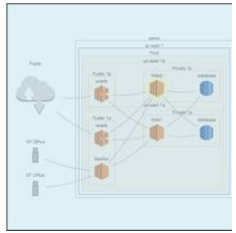
Neighbors:
-database
-Bastion
-database
-weblb
-weblb

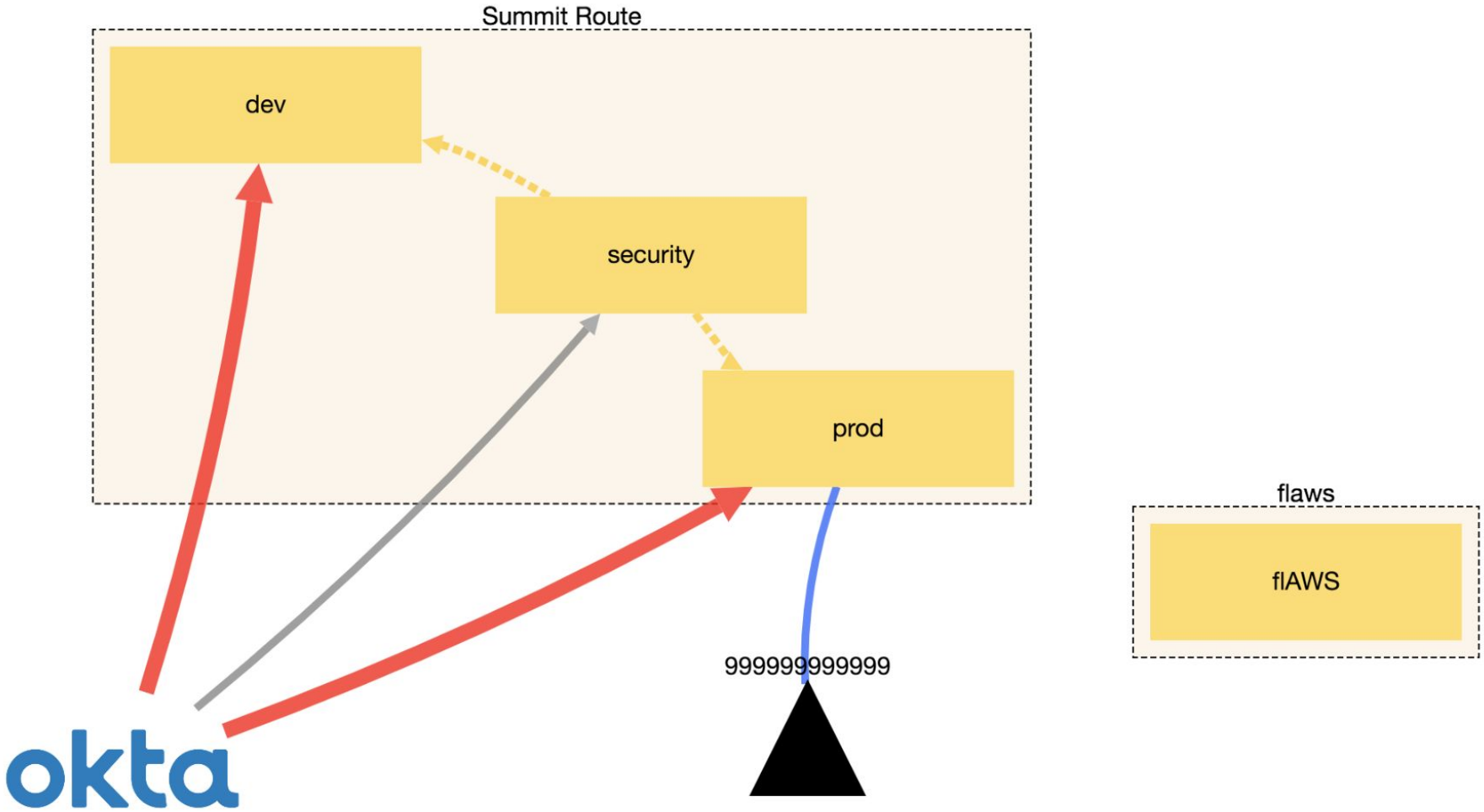
Siblings

Siblings:
-database

Children

Children:



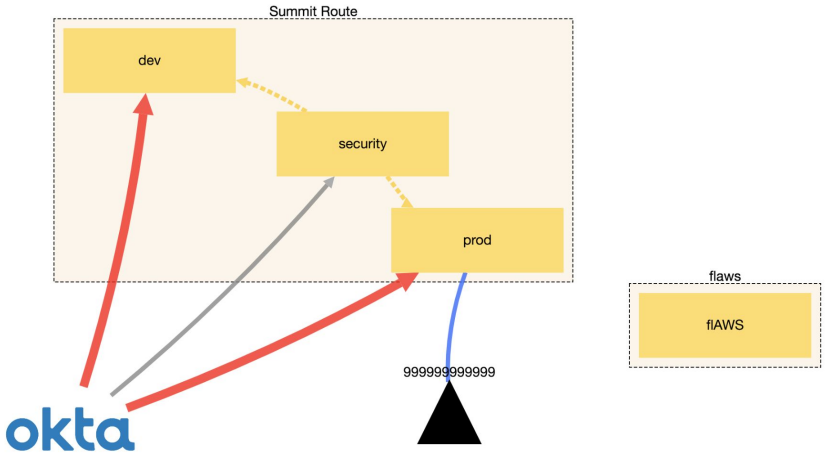

Summit Route
dev
security
prod
flaws
flAWS
999999999999
okta

How CloudMapper works - network view

1. collect
   a. Makes AWS API calls and stores all responses as json files locally.
2. prepare
   a. Identify nodes: Iterate hierarchies (account, region, VPC, Availability Zone, subnet) and find the resources within (EC2, RDS, ELB).
   b. Perform filtering.
   c. Identify edges: Iterate Security Groups to discover what can talk to what.
   d. Writes data.json
3. webserver
   a. Serves static files only -> Can be hosted as an S3 bucket or Github Pages.
   b. cytoscape.js loads the data.json and styling information
   c. Some glue code for loading additional plugins and displaying the info box.

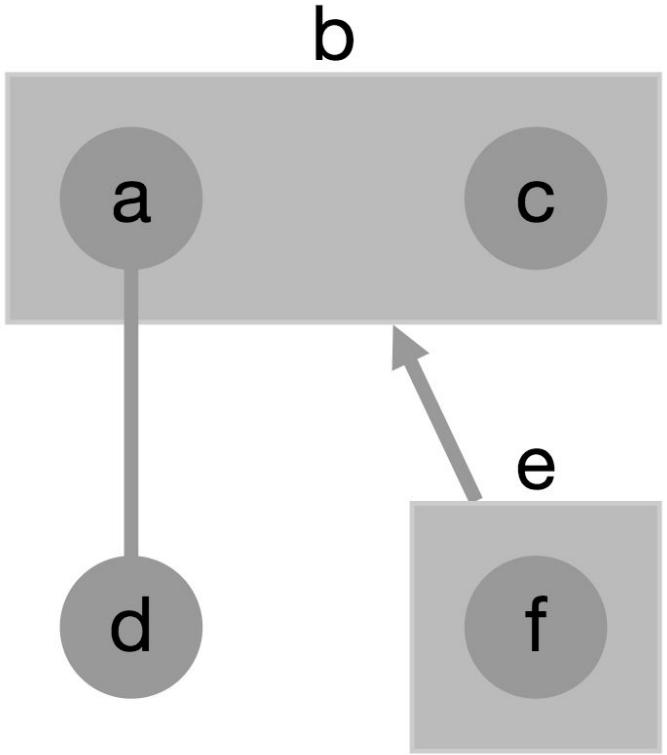
How CloudMapper works - web of trust view

1. wot - Same as network view, except
   a. Reviews access policies of resources (ex. S3 buckets).
   b. Reviews IAM policies for granted trust.
   c. Reviews network data for VPC peering.
   d. Does this across all AWS accounts of interest.
   e. Writes data.json



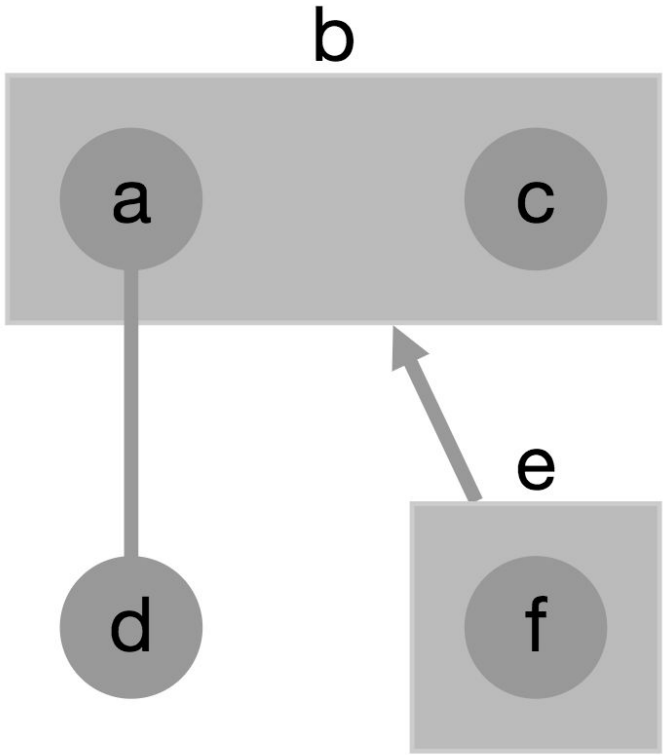
Why Cytoscape.js?

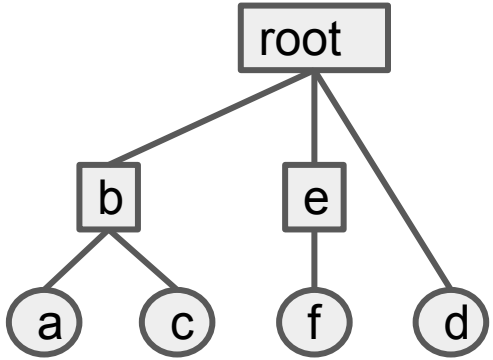
1. Free and open-source
2. Interactive in web-browser
3. Supports compound nodes and directed graphs with cycles
    - CoSE (Compound Spring Embedder) by the i-Vis Lab in Bilkent University[1,2]

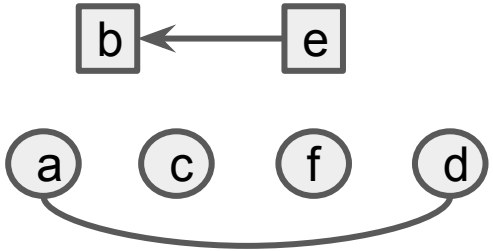


Matt Herman of Netflix compares Cytoscape.js to GraphViz[3]

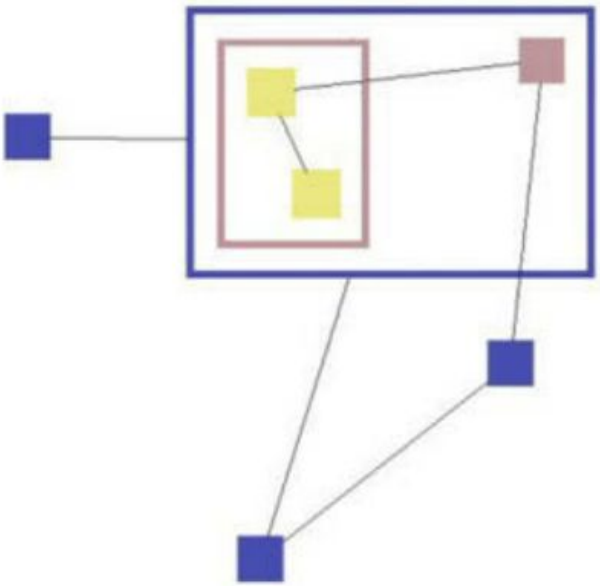
1. http://yoksis.bilkent.edu.tr/pdf/files/10.1016-j.ins.2008.11.017.pdf
2. https://github.com/cytoscape/cytoscape.js-cose-bilkent
3. @mpherman006 https://medium.com/@matt_herman/visualizing-attack-trees-c90f2b622ade

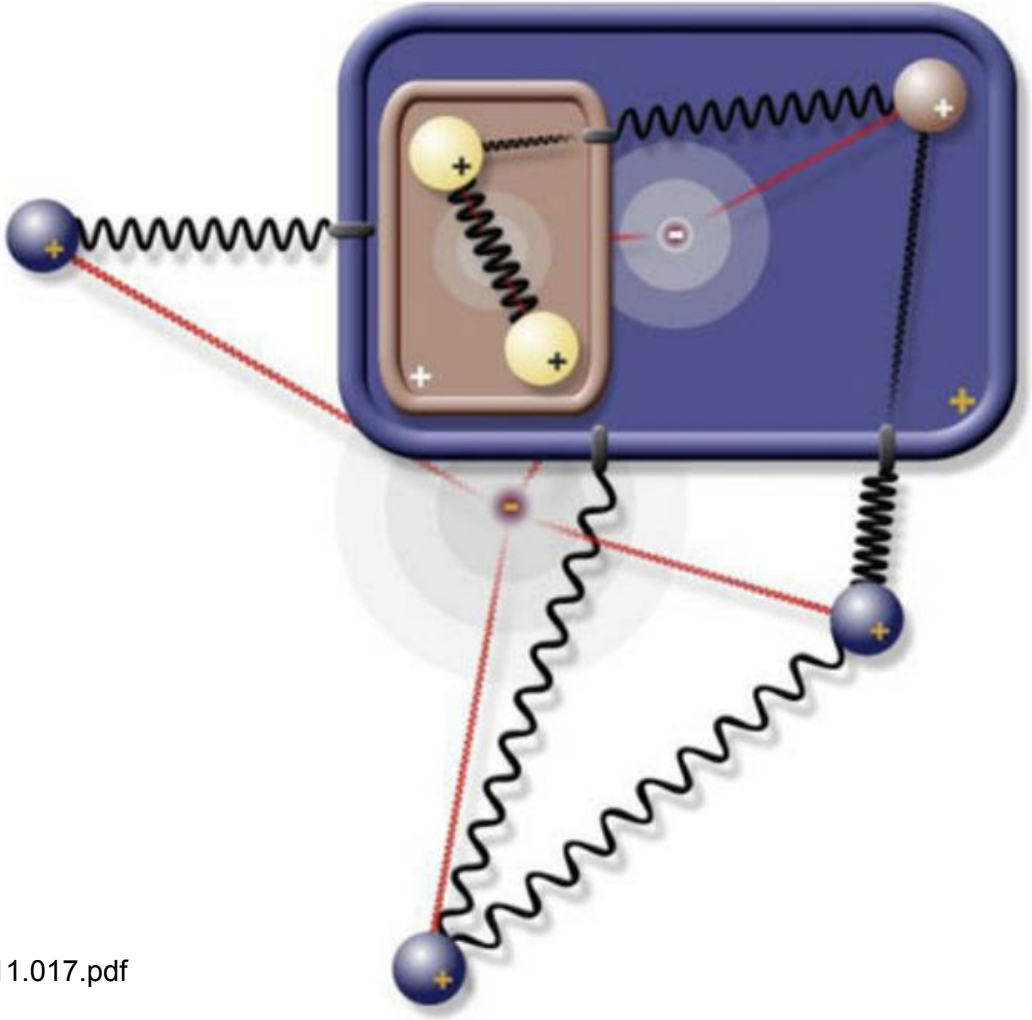






Compound graphs are two graphs in one.
They show parent-child relationships and connections.



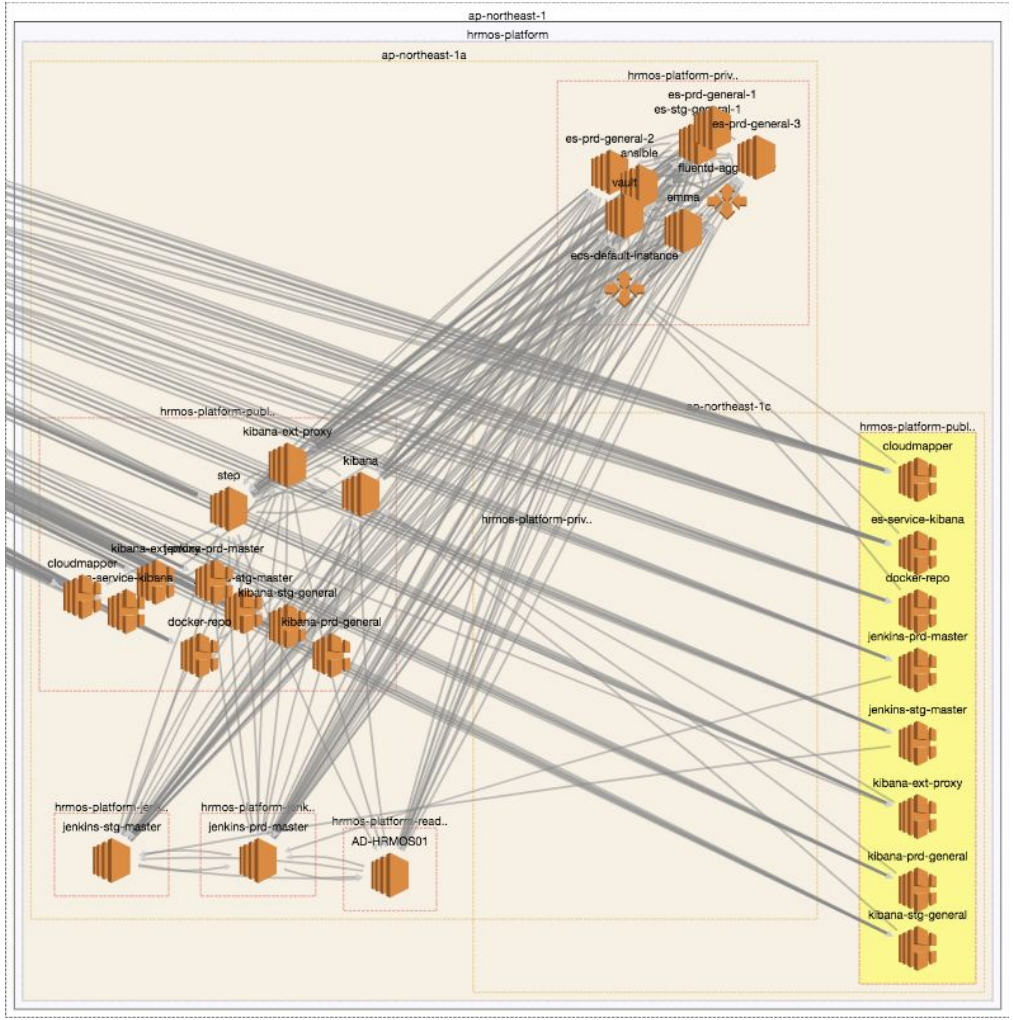


From http://yoksis.bilkent.edu.tr/pdf/files/10.1016-j.ins.2008.11.017.pdf

Problem: Having many nodes and edges is inherently complex.

Solution: Reduce what is displayed

- Different abstraction levels
  - Show a single subnet
  - Show all subnets collapsed to single nodes
- Reduce similar nodes to a single node
  - Autoscaling groups -> 1 node



Cytoscape.js limitations

- Limited in what you can draw
  - can't mix icons and shapes
  - changing sizes of nodes is awkward

How things are

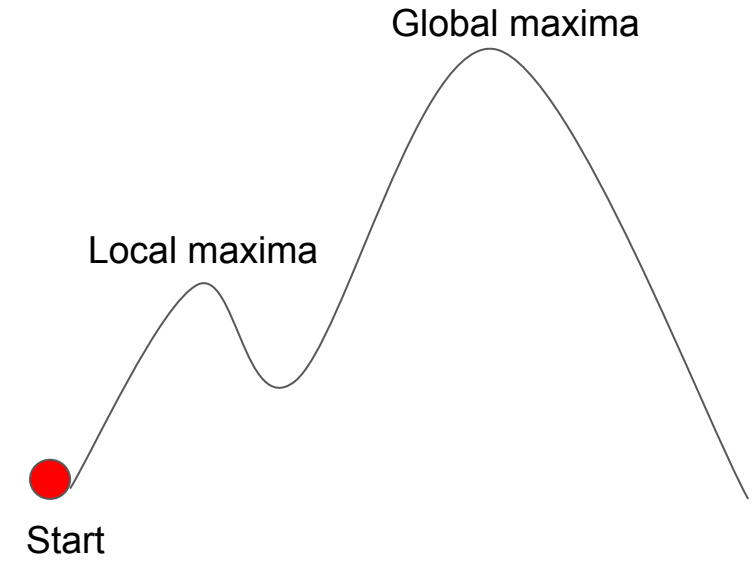
[Dev]
1 CPU, 1GB of RAM

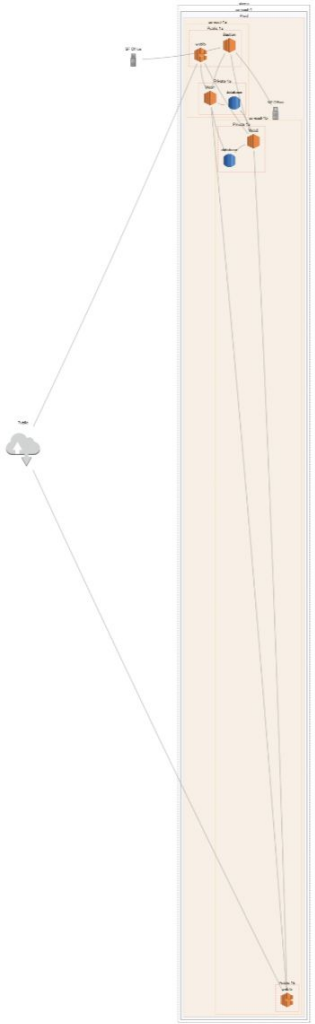
[Prod]
96 CPUs, 384GB of RAM

What I might want

Cytoscape.js limitations

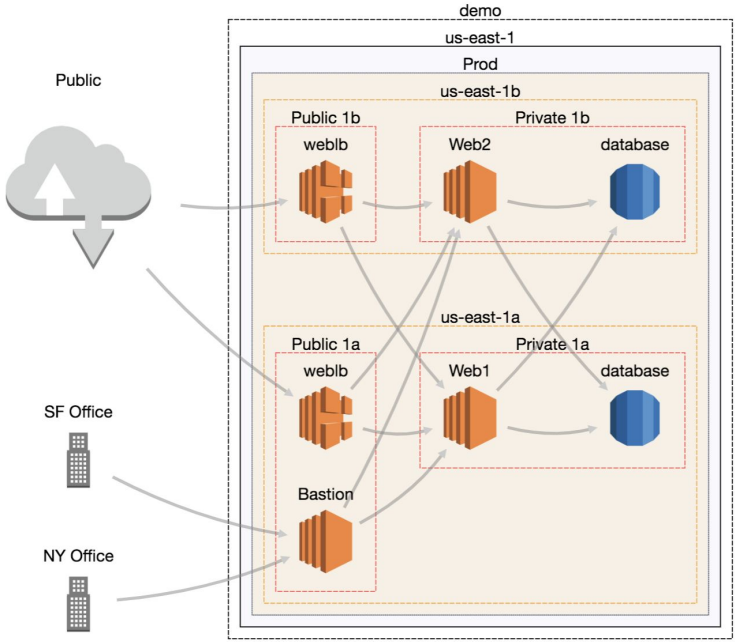
- Layout engine performs hill climbing, but without simulated annealing
  - Gets stuck in local maxima of the "best" view

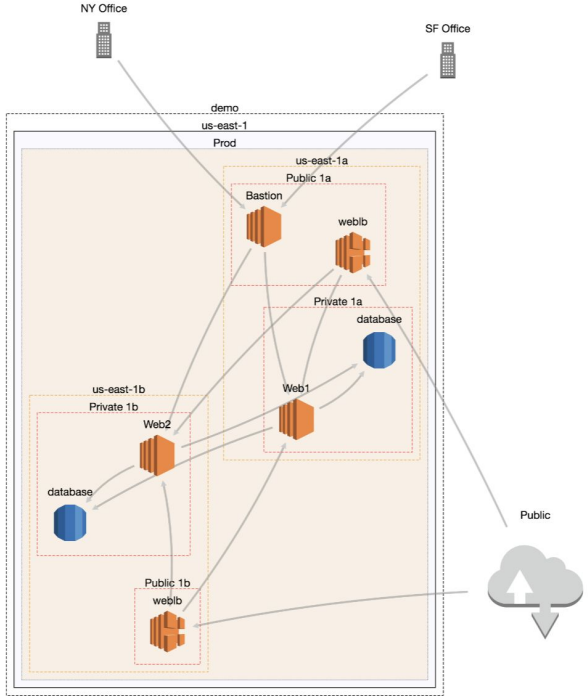




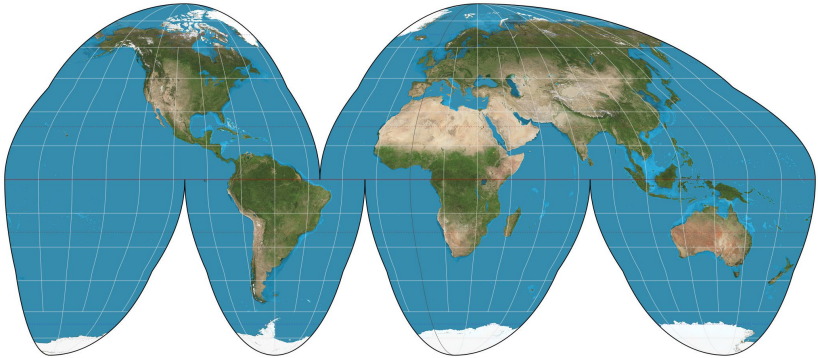
Cytoscape.js limitations

- Layout is based solely around node distances.
  - Does not take edge overlaps into consideration
  - No configurations for node affinity to certain locations (ex. always put "Internet" on left and databases on far right)





CloudMapper limitations



CloudMapper shows a model.

Models hide details: Sometimes important ones.

- Nodes are all represented as the same size.
  - A small EC2 looks the same as one that costs 1000x more
- Edges do not indicate what ports or protocols are open for communication.
- Edges do not show actual traffic, only what is possible.

CloudMapper makes a best effort

AWS can be very complex. CloudMapper can be wrong.

- No consideration of routing tables or NACLs.
- Some edge case capabilities of AWS are not considered.
- Simple heuristics are used to determine an admin IAM relationship vs normal.

Perfect is the enemy of good.

A visualization that may be incorrect in edge cases is better than no visualization.

Questions?

Key take-aways:

- Make best effort graphs: All graphs lack details and possibly miss rules.
- Reduce complexity by reducing information displayed.
- cytoscape.js is effective for interactive graphs, with limitations.

Scott Piper
@0xdabbad00
SummitRoute.com
scott@summitroute.com