



# Beyond Dashboards - Visualising Complex Systems

Andy Burgin - Wednesday 29<sup>th</sup> April 6PM

<https://www.meetup.com/Cloud-Native-Kubernetes-Manchester/>  
for sign up details.





# Beyond Dashboards

Visualising Complex Systems

# Hi, I'm Andy...

- Sky Betting and Gaming for the last 6 Years
- Lead Platform Engineer
- Pronouns He/Him



**LeedsDevops**  
Supporting the DevOps community in Leeds since 2013



# DevOps Enterprise Summit 2017 – John Allspaw

How Your Systems Keep Running Day After Day

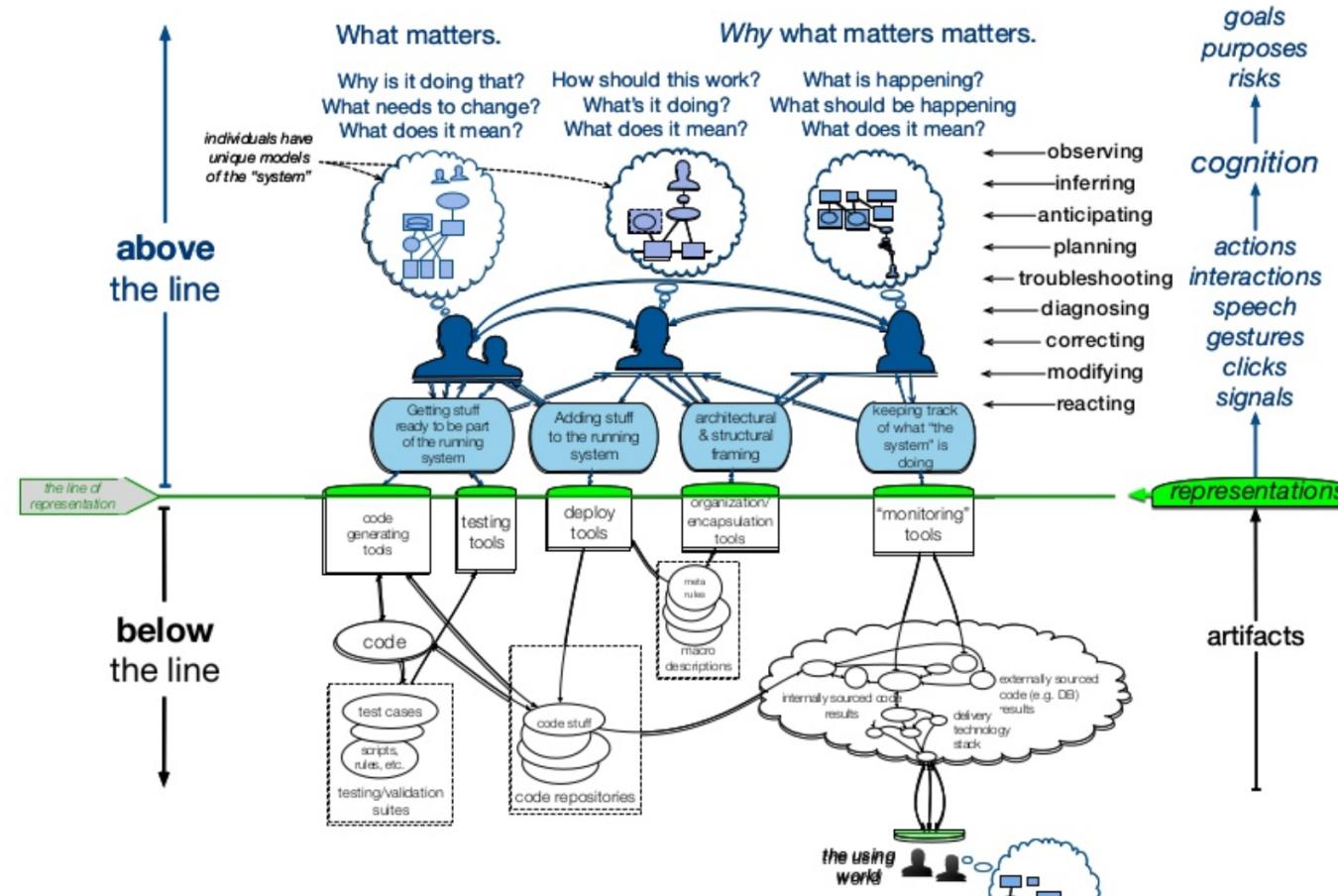


<https://www.youtube.com/watch?v=xA5U85LSk0M>

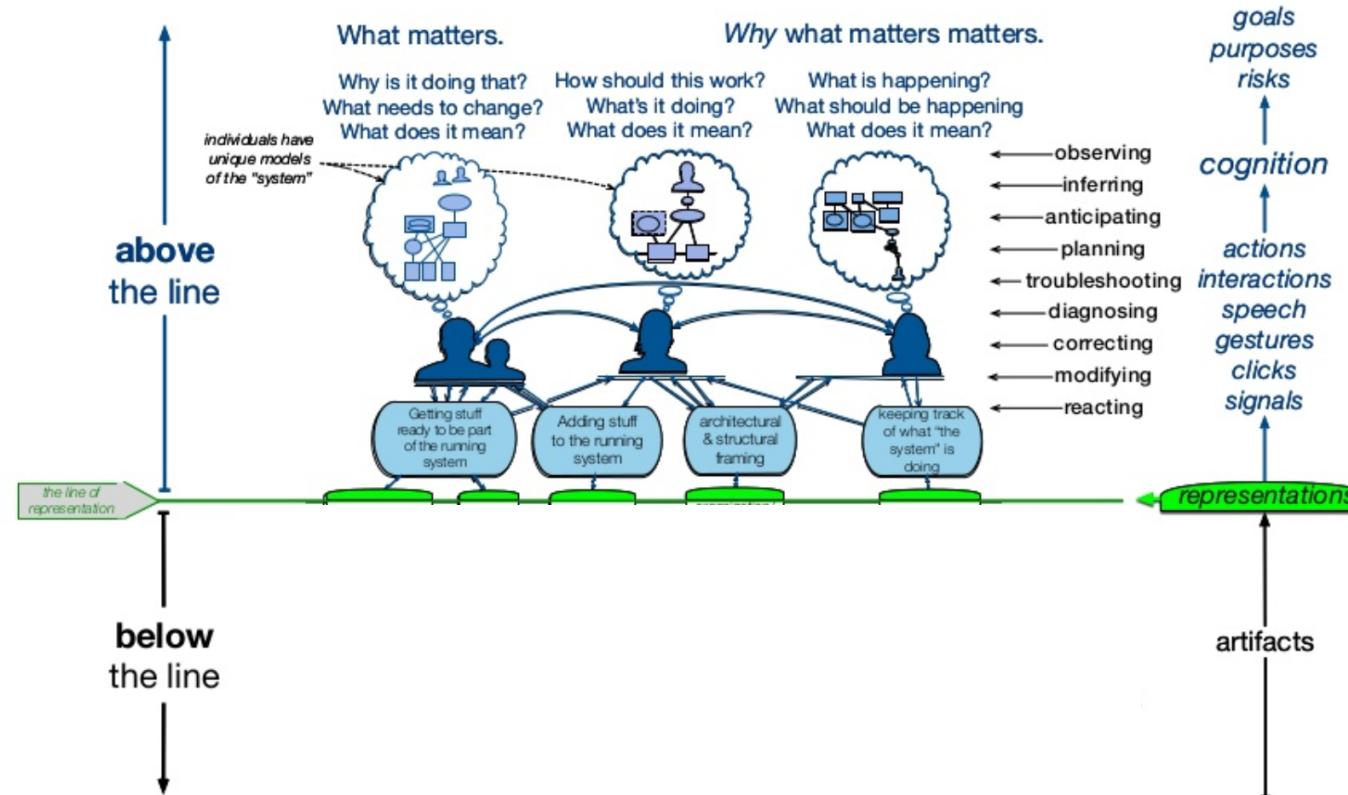
 @andyburgin



# DevOps Enterprise Summit 2017 – John Allspaw



# DevOps Enterprise Summit 2017 – John Allspaw



# DevOps Enterprise Summit 2017 – John Allspaw

Six themes were identified and discussed.

1 Capturing the value of anomalies through postmortems

2 Blame versus sanction in the aftermath of anomalies

3 Controlling the costs of coordination during anomaly response

4 Supporting work through improved visualizations

5 The strange loop quality of anomalies

6 Dark debt

**SN4U CATCHERS STELLA**  
Report from the SNAFUcatchers Workshop on Coping With Complexity  
Brooklyn NY, March 14-16, 2017

[Download PDF Version](#)

1. tl;dr and Executive Summary

- 1.1 tl;dr
- 1.2 Executive Summary

2. Introduction

- 2.1 About the SNAFUcatchers consortium and the STELLA meeting
- 2.2 The focus on handling anomalies
- 2.3 The above-the-line/below-the-line framework

3. Cases

- 3.1 Catching the Apache SNAFU
- 3.2 Catching the Travis CI SNAFU
- 3.3 Catching the Logstash SNAFU
- 3.4 Four cases
- 3.4.1 Features of the anomalies
- 3.4.2 Features of the anomaly responses
- 3.4.3 Guiding principles for uncertainty
- 3.4.4 The role of search
- 3.4.5 Evolutionary system representations
- 3.4.6 Generating hypotheses

Basic tools

- Coordination
- Communications in joint activity
- Shared artifacts
- The consequences of escalating consequences
- Managing risk
- Goal setting

3.5 Observations on the postmortem process

4. Themes

- 4.1 Capturing the value of anomalies
- 4.1.1 Technical issues in postmortems
- 4.1.2 Social issues in postmortems
- 4.2 Blame versus sanction in the aftermath of anomalies
- 4.3 Controlling the costs of coordination during anomaly response
- 4.3.1 Offloading work to low-tempo periods
- 4.3.2 Providing expertise on demand
- 4.3.3 Supporting communication and coordination with tools
- 4.4 Supporting anomaly response through improved visualizations
- 4.4.1 Understanding negative work in context of the starting point
- 4.5 Strange loops dependencies
- 4.6 Dark debt
- 4.6.1 Technical debt
- Origins of the debt metaphor
- Technical debt refactoring
- Technical debt 25 years on
- 4.6.2 Dark debt
- 5. Practical approaches for progress on coping with complexity
- 6. Back matter
- 6.1 Acknowledgements
- 6.2 Acknowledgements
- 6.3 Suggested citation for this report
- 7. References

**Winter storm STELLA**

Woods' Theorem: *As the complexity of a system increases, the accuracy of any single agent's own model of that system decreases rapidly.*

**1. tl;dr and Executive Summary**

**1.1 tl;dr**

A comprehensive workshop of high end techs reviewed postmortems to better understand how engineers cope with the complexity of anomalies (SNAFU and SNAFU catching episodes) and how to support them. These cases reveal common themes regarding factors that produce resilient performances. The themes that emerge also highlight opportunities to move forward.

**1.2 Executive Summary**

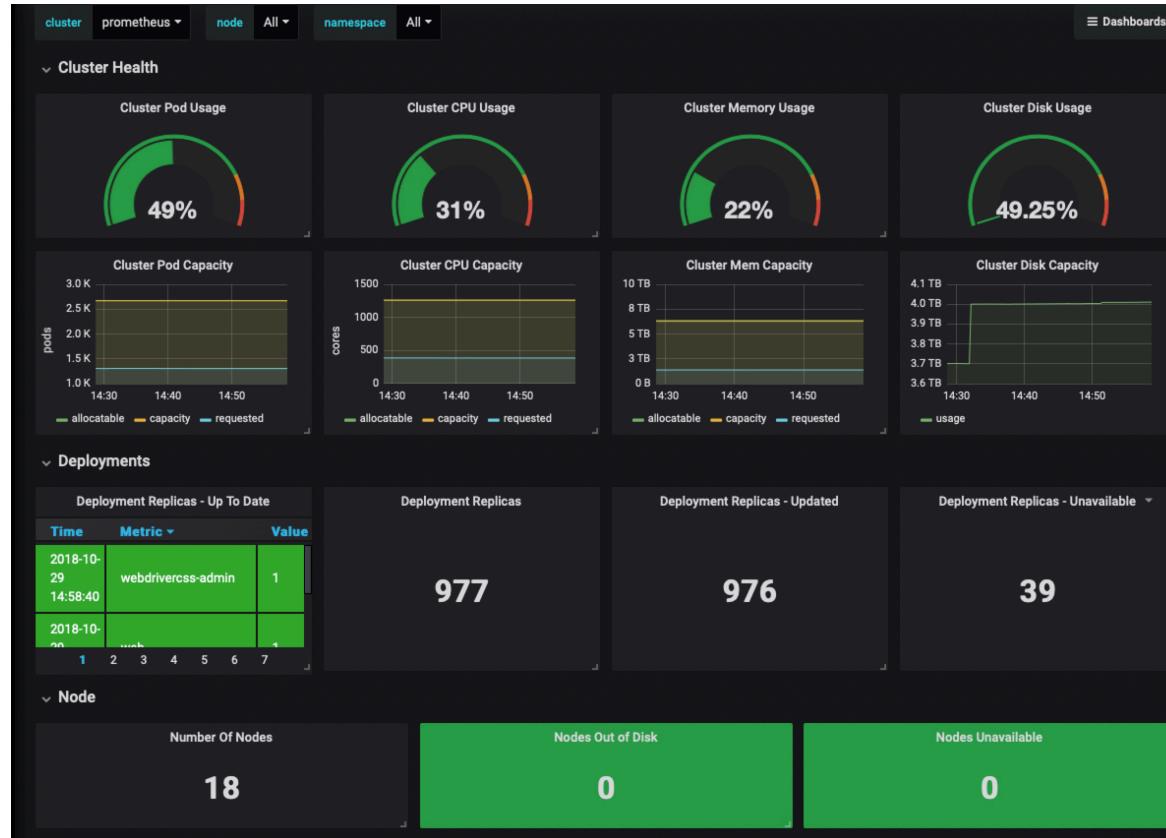
Current generation internet-facing technology platforms are complex and prone to brittle failure. Without the continuous effort of engineers to keep them running they would stop working -- many in days, most in weeks, all within a year. These platforms remain alive and functioning because workers are able to detect anomalies, diagnose their sources, remediate their effect, and repair their flaws and do so ceaselessly -- SNAFU Catching. Yet we know little about how they accomplish this vital work and...



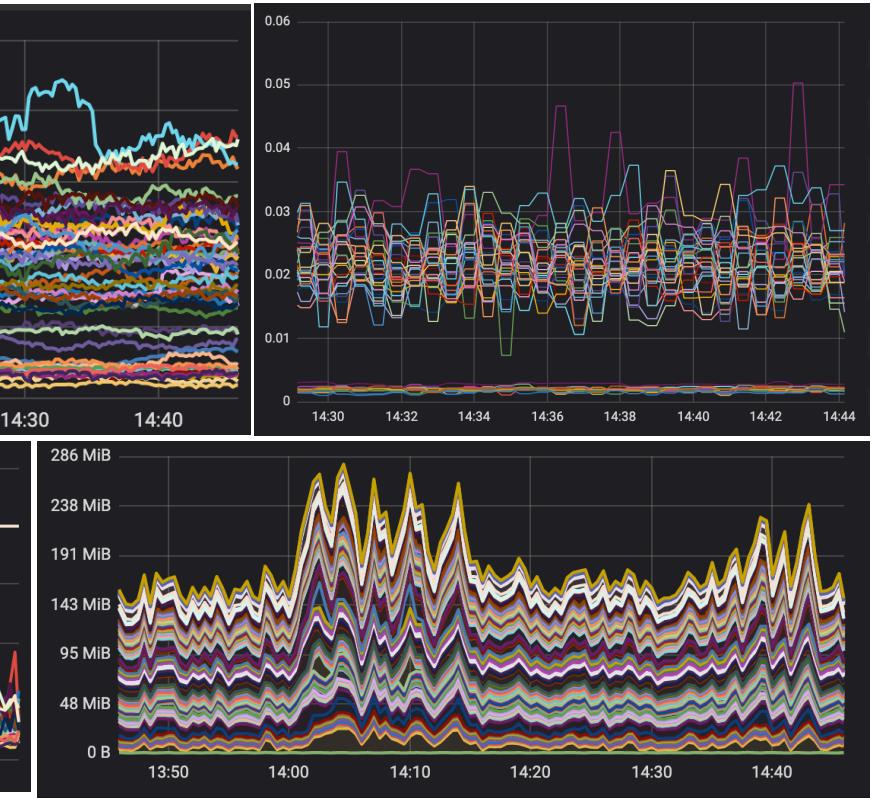
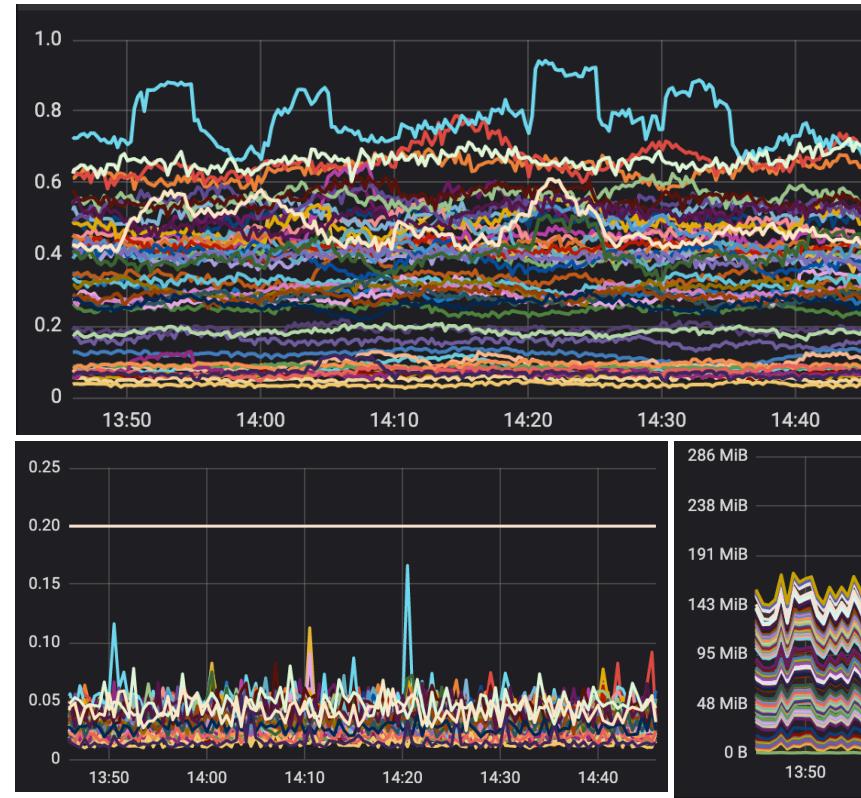
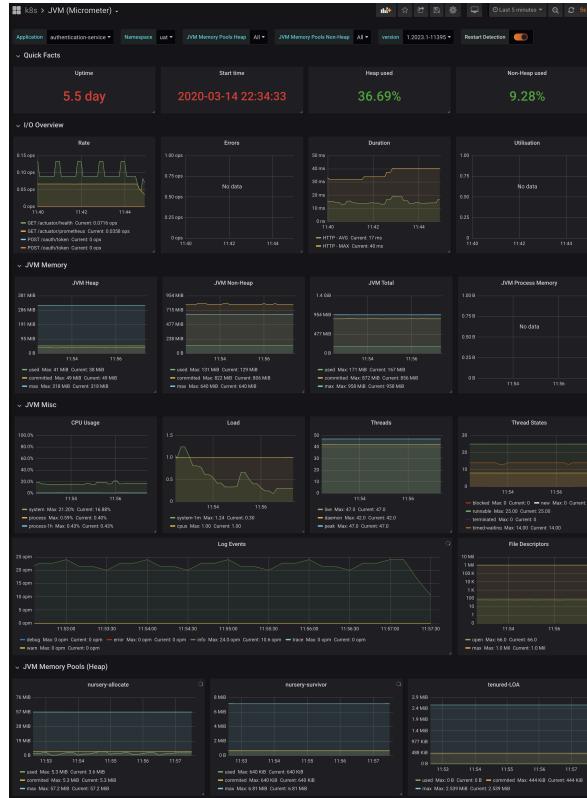
<http://stella.report>

# Visualising Complex Systems

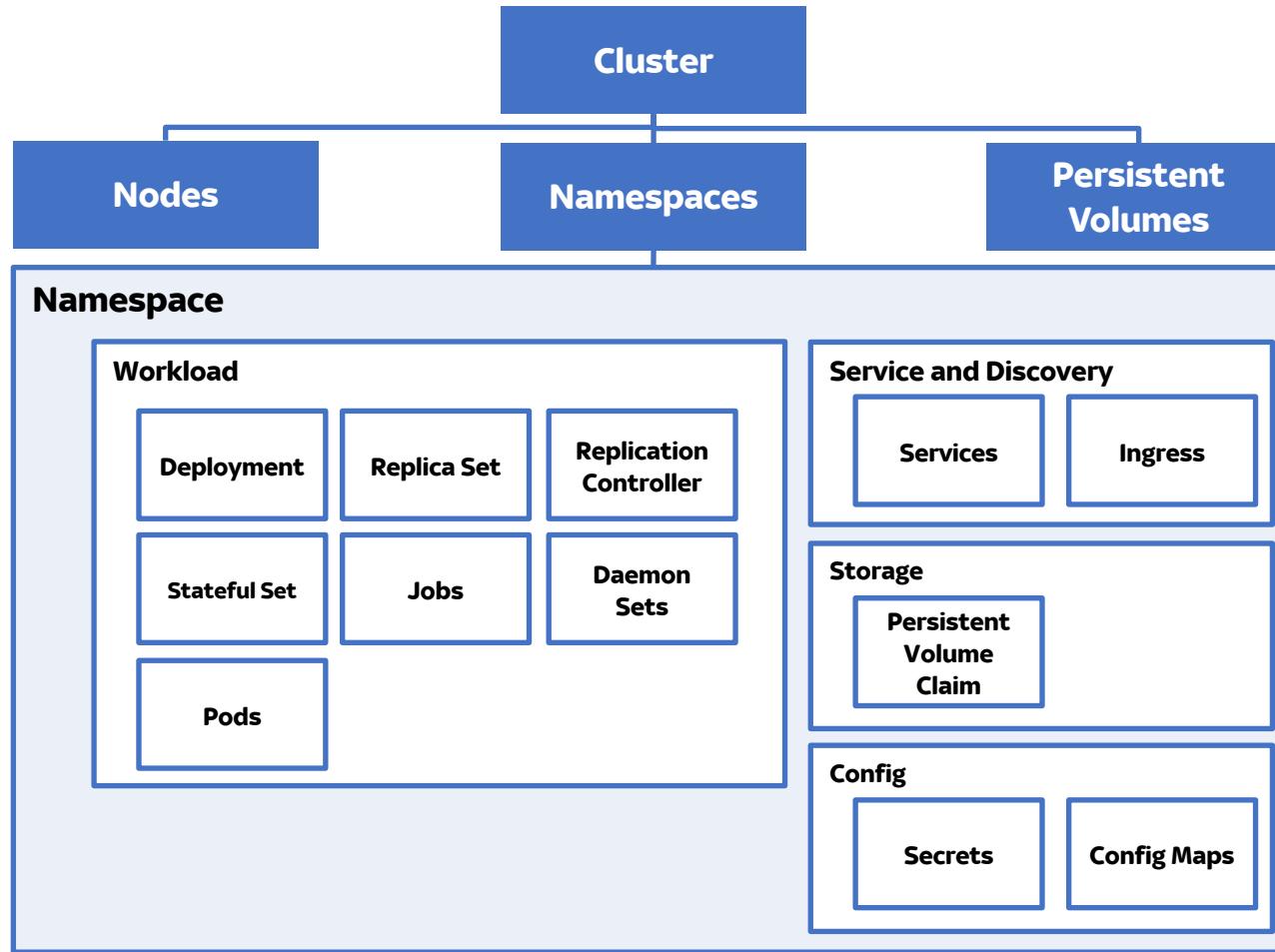
# What's Wrong With Dashboards ?



# What's Wrong With Dashboards ?



# Kubernetes Objects

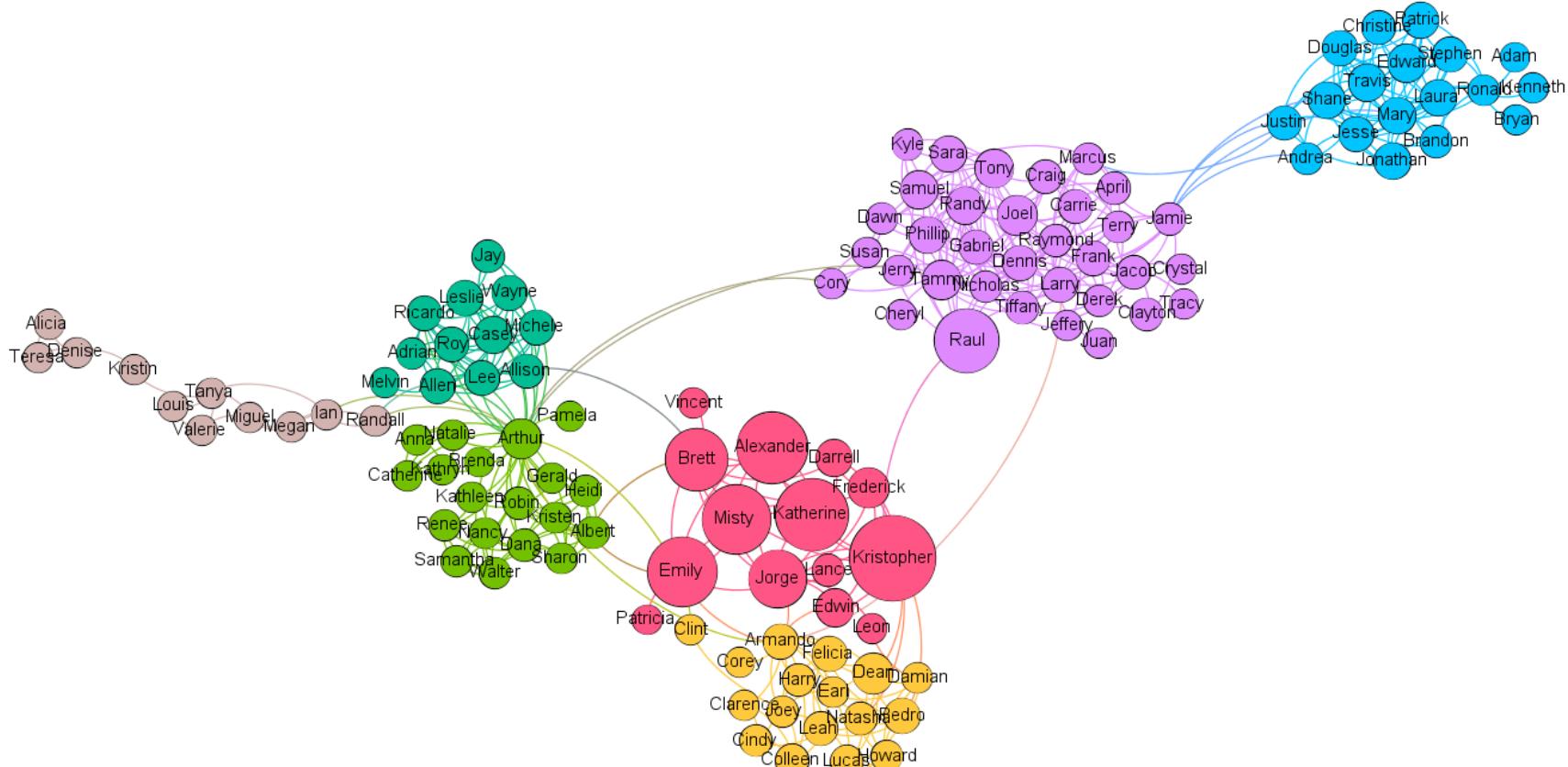


## What Do Your Clusters Look Like ?

**kubectl get all -A**

# Graphs

# Graphs



# Graphs



# Graphs

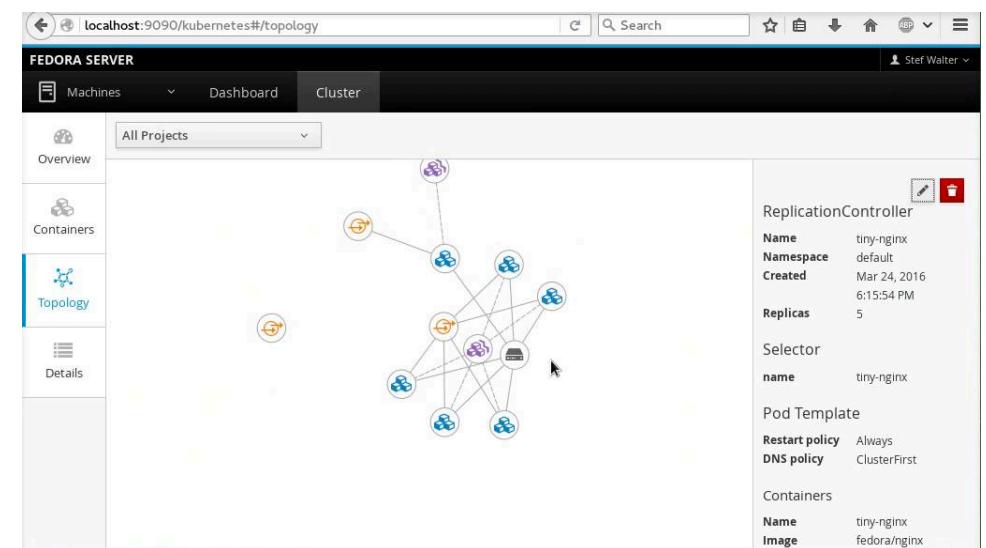
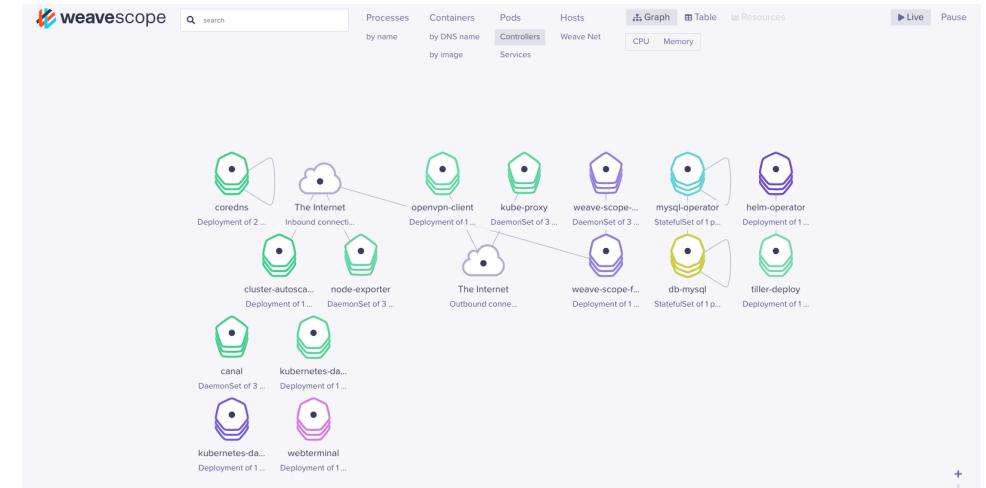
Nodes:

- Person
- Class/Object
- Device
- Port/Socket
- Record/Row
- Services/APis

Edges:

- Like/Dislike
- Relationship
- Usage
- Weight/Volume
- Link
- Dependencies

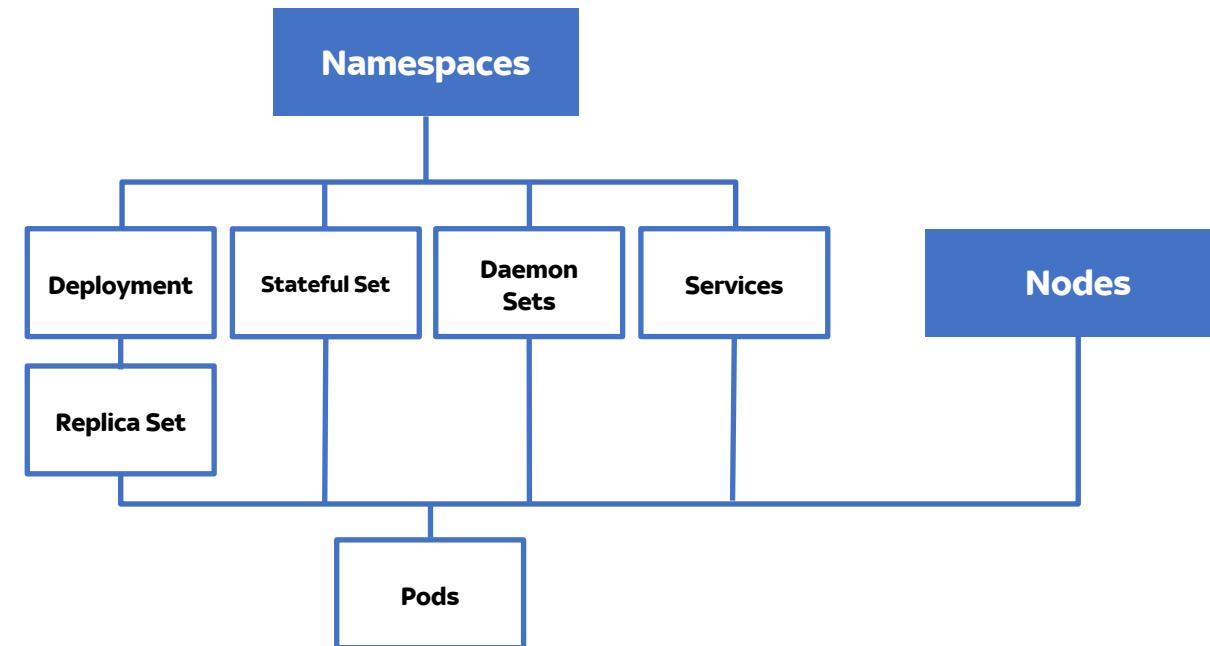
# Graphs



# Demo

Warning Contains Moving Images

# Kubernetes Objects



```
$ docker run -p 7473:7473 -p 7474:7474 -p 7687:7687 -p 6000:6000 -v $HOME/neo4j/data:/data -v $HOME/neo4j/plugins:/plugins --name=neo4j -e NEO4J_apoc_export_file_enabled=true -e NEO4J_apoc_import_file_enabled=true -e NEO4J_apoc_import_file_use_neo4j_config_bolt_tls_level=OPTIONAL neo4j:3.4.12
Active database: graph.db
Directories in use:
  home:          /var/lib/neo4j
  config:        /var/lib/neo4j/conf
  logs:          /var/lib/neo4j/logs
  plugins:       /plugins
  import:         /var/lib/neo4j/import
  data:          /var/lib/neo4j/data
  certificates: /var/lib/neo4j/certificates
  run:           /var/lib/neo4j/run
Starting Neo4j.
2020-04-10 13:13:30.180+0000 WARN  Unknown config option: causal_clustering.discovery_listen_address
2020-04-10 13:13:30.187+0000 WARN  Unknown config option: causal_clustering.raft_advertised_address
2020-04-10 13:13:30.188+0000 WARN  Unknown config option: causal_clustering.raft_listen_address
2020-04-10 13:13:30.188+0000 WARN  Unknown config option: ha.host.coordination
2020-04-10 13:13:30.189+0000 WARN  Unknown config option: causal_clustering.transaction_advertised_address
2020-04-10 13:13:30.190+0000 WARN  Unknown config option: causal_clustering.discovery_advertised_address
2020-04-10 13:13:30.191+0000 WARN  Unknown config option: ha.host.data
2020-04-10 13:13:30.191+0000 WARN  Unknown config option: causal_clustering.transaction_listen_address
2020-04-10 13:13:30.234+0000 INFO  ===== Neo4j 3.4.12 =====
2020-04-10 13:13:30.312+0000 INFO  Starting...
2020-04-10 13:13:42.132+0000 INFO  Bolt enabled on 0.0.0.0:7687.
2020-04-10 13:14:04.384+0000 INFO  Started.
2020-04-10 13:14:07.612+0000 INFO  Remote interface available at http://localhost:7474/
```

main.go — k8stest

R... main.go × index.html go.mod

VARIABLES

```
main.go > {} main > main
178     fmt.Fprintf(os.Stderr, "error: %v\n", err)
179     os.Exit(2)
180 }
181 // loop pod
182 for _, pod := range pods.Items {
183     fmt.Fprintf(os.Stdout, "    Pod : %v %v\n", pod.Name, pod.UID)
184     err = neo4jAddK8sObj(neo4jSession, &pod, k8sClient)
185     if err != nil {
186         fmt.Fprintf(os.Stderr, "error: %v\n", err)
187         os.Exit(2)
188     }
189     podCount++
190     if controllerRef := metav1.GetControllerOf(&pod); controllerRef != nil {
191         fmt.Fprintf(os.Stdout, "    xxxx pod: %v %v %v\n", pod.Name, controllerRef.UID,
192                     // neo4j add link pod to parent
193                     neo4jLink(neo4jSession, string(controllerRef.Kind), string(controllerRef.UID),
194                     } else {
195             fmt.Fprintf(os.Stdout, "    pod: %v linked to namespace %v\n", pod.Name, names
196             // neo4j link pod to namespace
197             neo4jLink(neo4jSession, "Namespace", string(namespace.UID), "Pod", string(pod.U
198         }
199     //link node to pod
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 5

☰ ^ X



BREAKPOINTS



localhost:800 x | M Using Neo4J x | bolt://localhost x | Force-Directed x | Visualizing G x | Hive Plots - L x | Hive Plots - L x | Gephi - The C x | +

localhost:7474/browser/

Favorites

Saved Scripts +New Folder

- MATCH (p:Pod) RETURN p.cpuLimitTotal, p.name, p.namespace ORDER BY
- MATCH (n:Namespace) -[\*]-> (p:Pod) RETURN n.name, n.created,
- MATCH s = (d:Deployment) -[\*]-> (p:Pod) WHERE d.name="trickster" RETURN
- MATCH s = (n:Namespace) -[\*]-> (p:Pod) WHERE p.name="trickster-

gephi

utility

Sample Scripts

Basic Queries

Example Graphs

Data Profiling

Common Procedures

\$

\$ MATCH s = (d:Deployment) -[\*]-> (p:Pod) WHERE d.name="trickster" RETURN...

Graph Table Text Code

\*(10) Deployment(1) ReplicaSet(3) Pod(2) Namespace(1) Service(1) Node(2)

\*(11) OWNS(11)

Pod <id>: 122286 age: 14837 cpulimittotal: 800 cpurequesttotal: 800 created: 2020-03-23 11:25:49 +0000 GMT



To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#),  
including cookie policy.

[Sign in](#)[Get started](#)

ANALYTICS

CLOUD

VISUALIZATION

HOW-TO

IMPORT/EXPORT

ANNOUNCEMENTS

DISCUSSIONS

DEVELOPER GUIDES

# Visualizing Graphs in 3D with WebGL

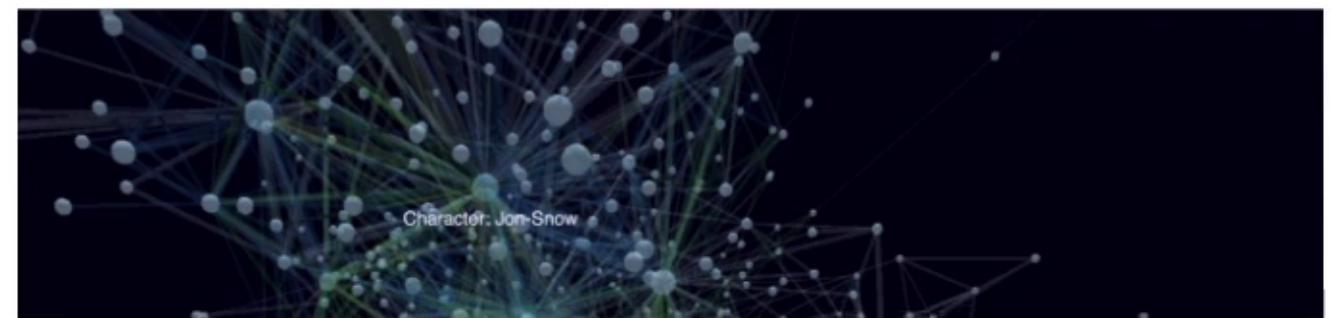
While looking for efficient graph visualization libraries for large scale rendering, I came across [3d-force-graph](#), a really neat wrapper around [three.js](#) for graph visualization. Check out that repository after reading this, they have many more examples and settings to explore.



Michael Hunger

[Follow](#)

Jul 23, 2018 · 6 min read





Left-click: rotate, Mouse wheel/middle-click: zoom, Right-click: pan

Martin Krzywinski // Circos / Genome Paths / Genome Informatics 2010 / Presidential Debates / HDTR / Schemaball / 4ness of π / GSC 10th / clock / photography / spam poetry / ascii / LOTRO



## HIVE PLOTS

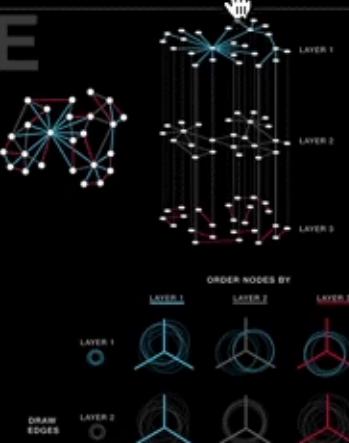
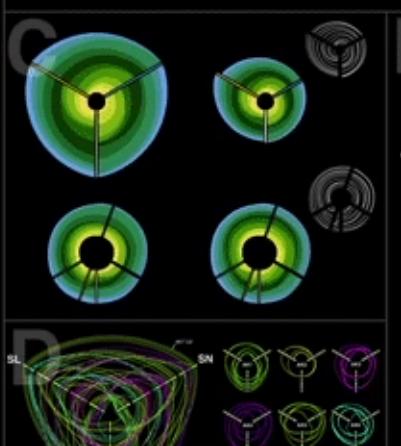
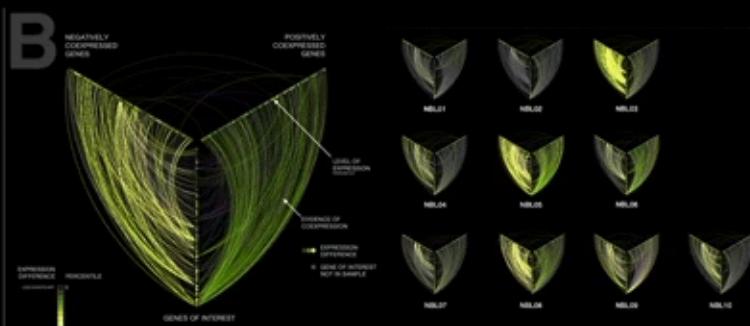
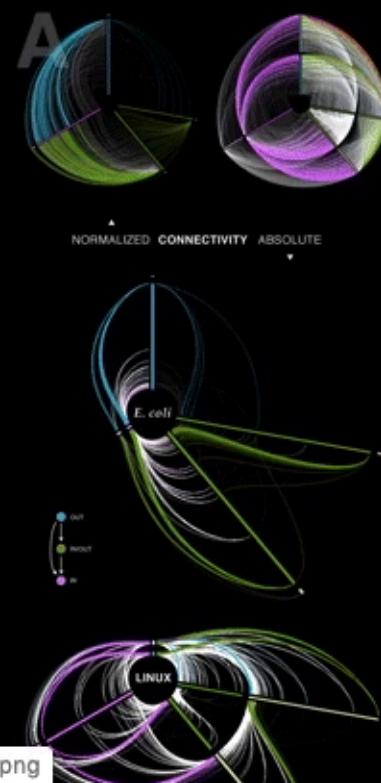
### RATIONAL NETWORK VISUALIZATION — FAREWELL TO HAIRBALLS

Martin Krzywinski, Genome Sciences Center, Vancouver, BC



#### PUBLISHED IN BRIEFINGS IN BIOINFORMATICS

Krzywinski M, Birol I, Jones S, Marra M (2011). [Hive Plots — Rational Approach to Visualizing Networks](#). *Briefings in Bioinformatics* (early access 9 December 2011, doi: 10.1093/bib/bbr069). ([download citation](#))



THE HIVE PLOT IS A PERCEPTUALLY UNIFORM AND SCALABLE LINEAR LAYOUT VISUALIZATION FOR NETWORK VISUAL ANALYTICS

UNDERSTANDING NETWORK STRUCTURE WITH HIVE PLOTS.  
(A) Normalized (top) and absolute (bottom) connectivity of *E. coli* gene regulatory network and Linux function call network (Yan *et al.*)

(B) Gene co-regulation networks in neuroblastoma samples.

(C) Network edges shown as ribbons creating circularly composited stacked bar plots (a periodic streamgraph).

(D) Syntenic network of three modern crucifer species to ancestral genome.

(E) Layered network correlation matrix. In each cell two layers  $u, v$  are depicted with  $u$  used to order axes and nodes while links for  $v$  are shown.

ZOOM GET SLIDES

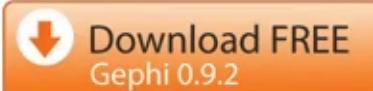
[Download](#) [Blog](#) [Wiki](#) [Forum](#) [Support](#) [Bug tracker](#)[Home](#) [Features](#) [Learn](#) [Develop](#) [Plugins](#) [Services](#) [Consortium](#)

# The Open Graph Viz Platform

Gephi is the leading visualization and exploration software for all kinds of graphs and networks. Gephi is open-source and free.

Runs on Windows, Mac OS X and Linux.

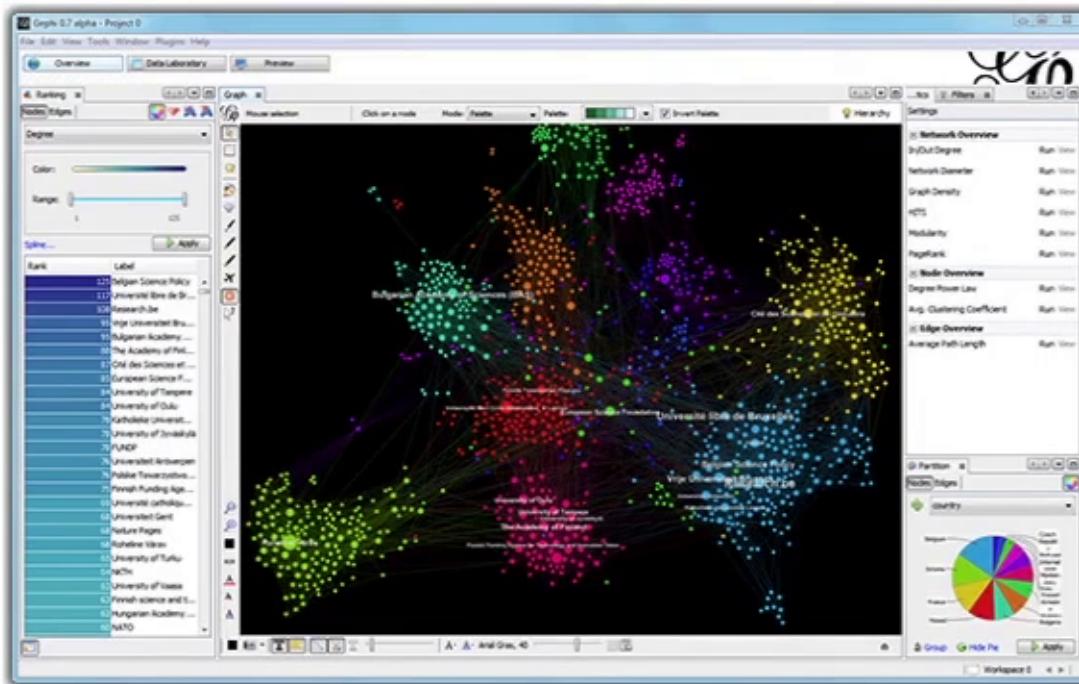
[Learn More on Gephi Platform >](#)



[Release Notes](#) | [System Requirements](#)

► [Features](#)  
► [Quick start](#)

► [Screenshots](#)  
► [Videos](#)



Support us! We are non-profit. Help us to innovate and empower the community by donating only 8€:

[Donate](#)



localhost:7474/browser/

**Favorites**

**Saved Scripts** +New Folder

- MATCH (p:Pod) RETURN p.cpuLimitTotal, p.name, p.namespace ORDER BY
- MATCH (n:Namespace) -[\*]-> (p:Pod) RETURN n.name, n.created,
- MATCH s = (d:Deployment) -[\*]-> (p:Pod) WHERE d.name="trickster" RETURN
- MATCH s = (n:Namespace) -[\*]-> (p:Pod) WHERE p.name="trickster"

**gephi**

- MATCH path = (n)-[]->() CALL apoc.gephi.add('host.docker')
- MATCH path = (n) WHERE not((n)-[]-()) CALL apoc.gephi.add('host.docker')

**utility**

**Sample Scripts**

```
1 MATCH path I= (n)-[]->()
2 CALL apoc.gephi.add('host.docker.internal',
  'workspace1',path,'weight',
  ["age","completions","cpulimittotal","cpurequesttotal","created","defcpu",
  "defmem","defreqcpu","defreqmem","hostip","hostname","internaldns",
  "internalip","limitcount","memlimittotal","memrequesttotal",
  "name","namespace","node","nodealloccpu","nodeallocmem","nodecancpu"])
```

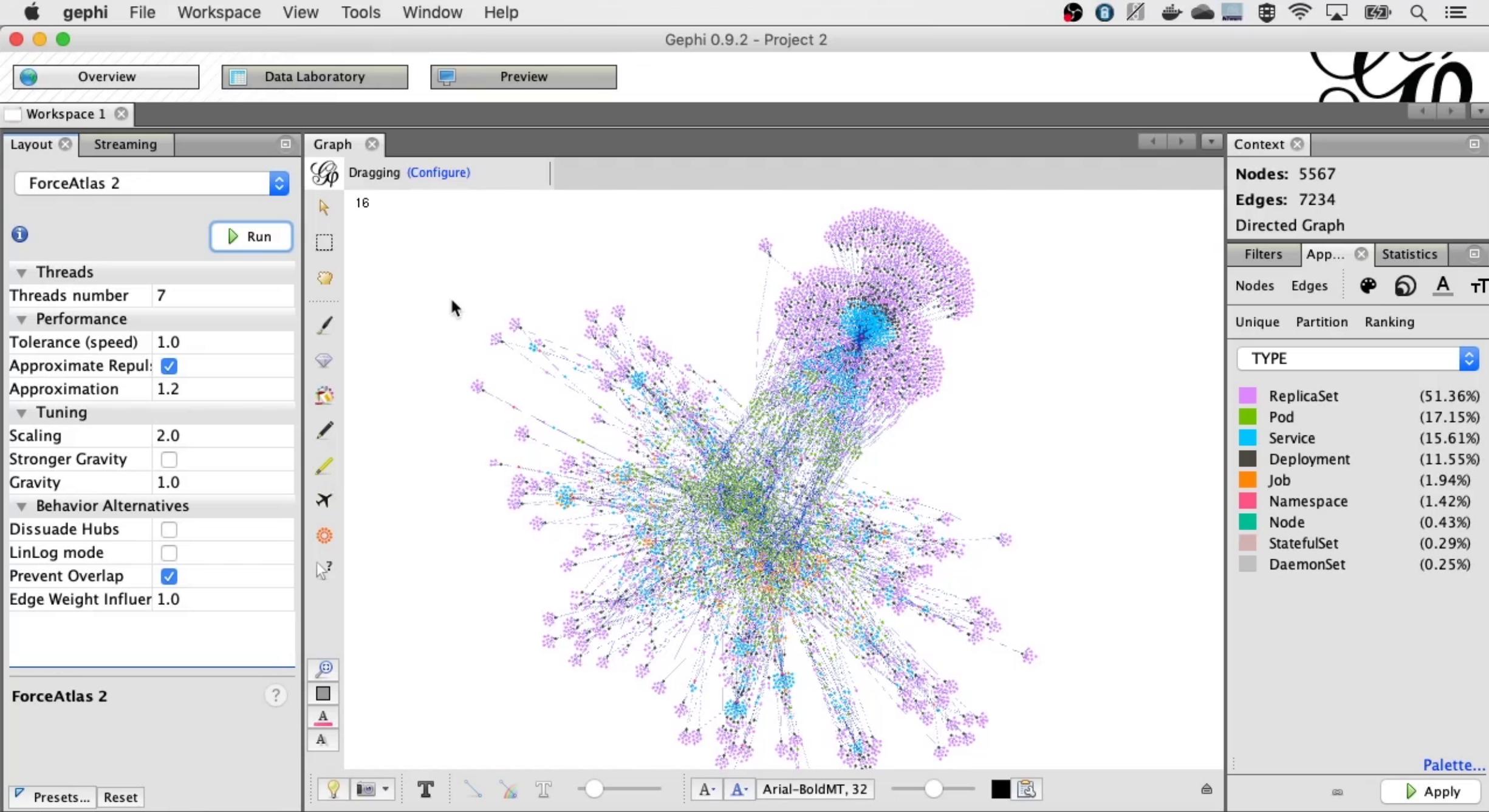
\$ MATCH s = (n:Namespace) -[\*]-> (p:Pod) WHERE p.name="trickster-69db6b8..."

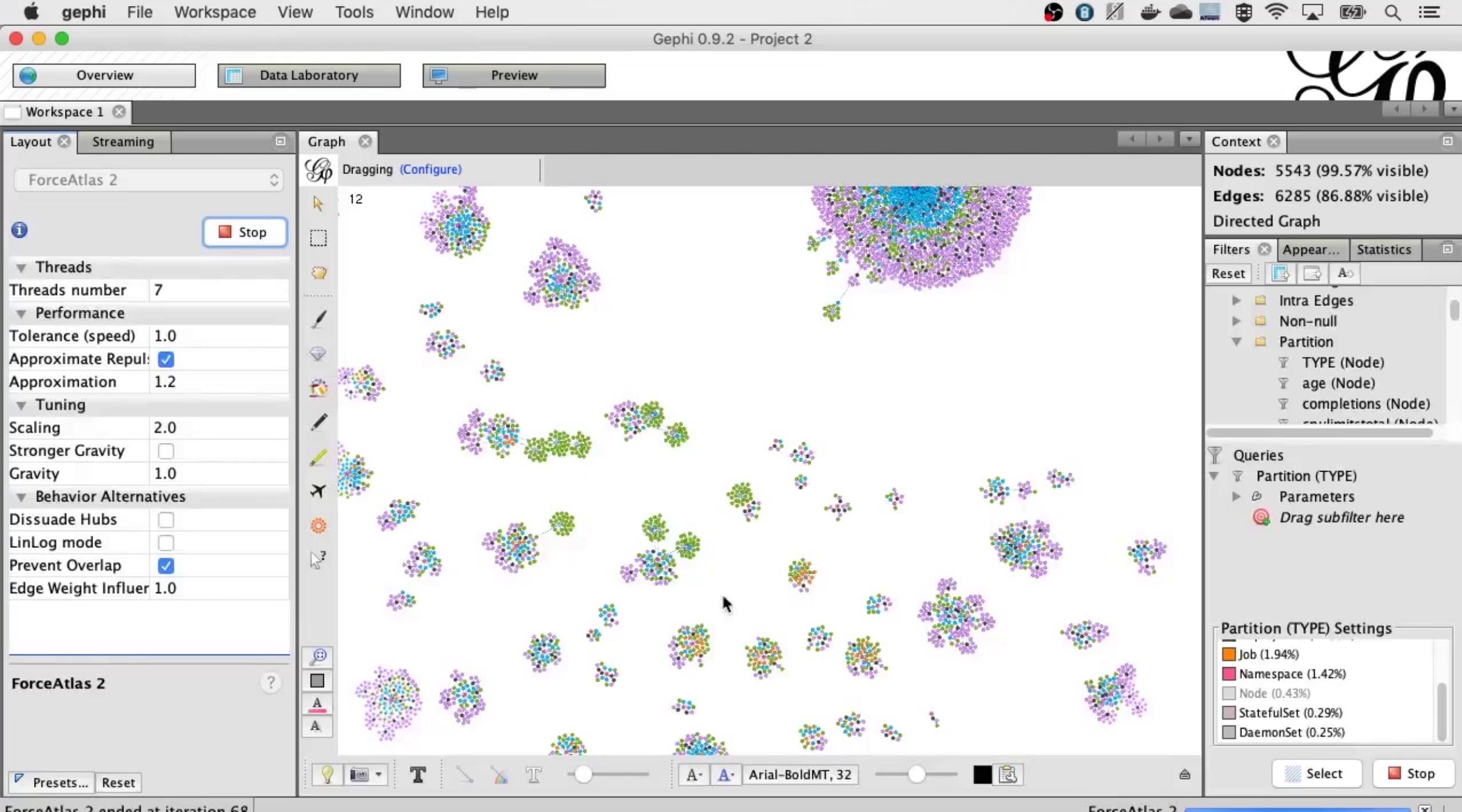
Graph Table Text Code

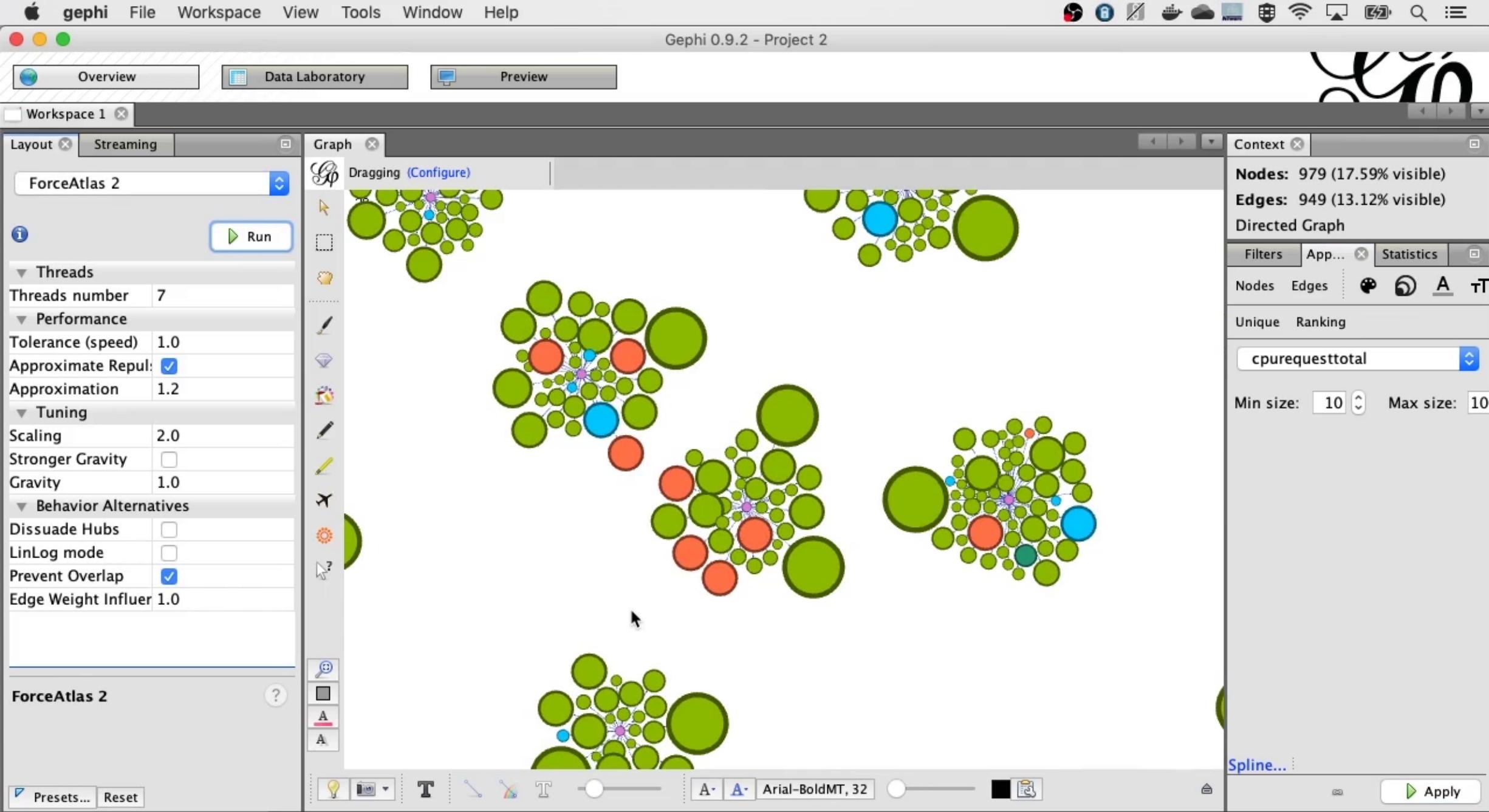
\*(51) Namespace(1) Deployment(7) ReplicaSet(16) Pod(7) Service(13) Node(2) DaemonSet(2)

\*(61) OWNS(61)

The graph displays a complex network of relationships between Kubernetes resources. Nodes are categorized by color: green for namespaces, orange for deployment replicasets, red for pods, blue for services, brown for nodes, and grey for daemonsets. Edges connect nodes based on their roles and ownership. For example, a 'monitoring' service node is shown with multiple edges to various pod nodes, indicating it owns them. Other edges show relationships like 'Owning Container' and 'Owning Pod' between different components.







recorder.go — Untitled (Workspace)

RUN La

VARIABLES

WATCH

CALL STACK

BREAKPOINTS

record.go x main.go recorder.log recorder.gexf

```
k8srecorder > recorder.go > {} main > main
129         os.Exit(1)
130     }
131
132     informerFactory := informers.NewSharedInformerFactory(clientset, time.Second*30)
133
134     informerFactory.Core().V1().Nodes().Informer().AddEventHandler(cache.ResourceEventHandler{
135         AddFunc:    addObject,
136         DeleteFunc: deleteObject,
137     })
138
139     informerFactory.Core().V1().Namespaces().Informer().AddEventHandler(cache.ResourceEventHandler{
140         AddFunc:    addObject,
141         DeleteFunc: deleteObject,
142     })
143
144     informerFactory.Apps().V1().DaemonSets().Informer().AddEventHandler(cache.ResourceEventHandler{
145         AddFunc:    addObject,
146         DeleteFunc: deleteObject,
147     })
148
149     informerFactory.Apps().V1().StatefulSets().Informer().AddEventHandler(cache.ResourceEventHandler{
150         AddFunc:    addObject,
151         DeleteFunc: deleteObject,
152     })
153
154     informerFactory.Batch().V1().Jobs().Informer().AddEventHandler(cache.ResourceEventHandler{
155         AddFunc:    addObject,
```

Ln 134, Col 32 (1 selected) Tab Size: 4 UTF-8 LF Go Analysis Tools Missing

RUN ▶ La ▾ ⚙️ ⌂ recorder.go main.go recorder.log recorder.gexf

VARIABLES k8srecord2gexf > recorder.log

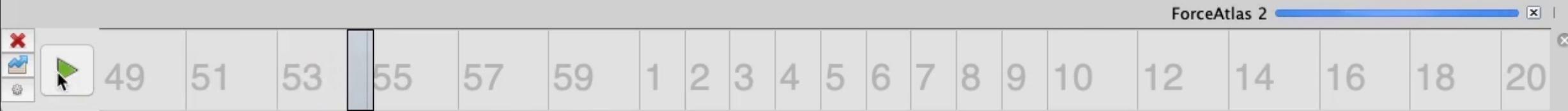
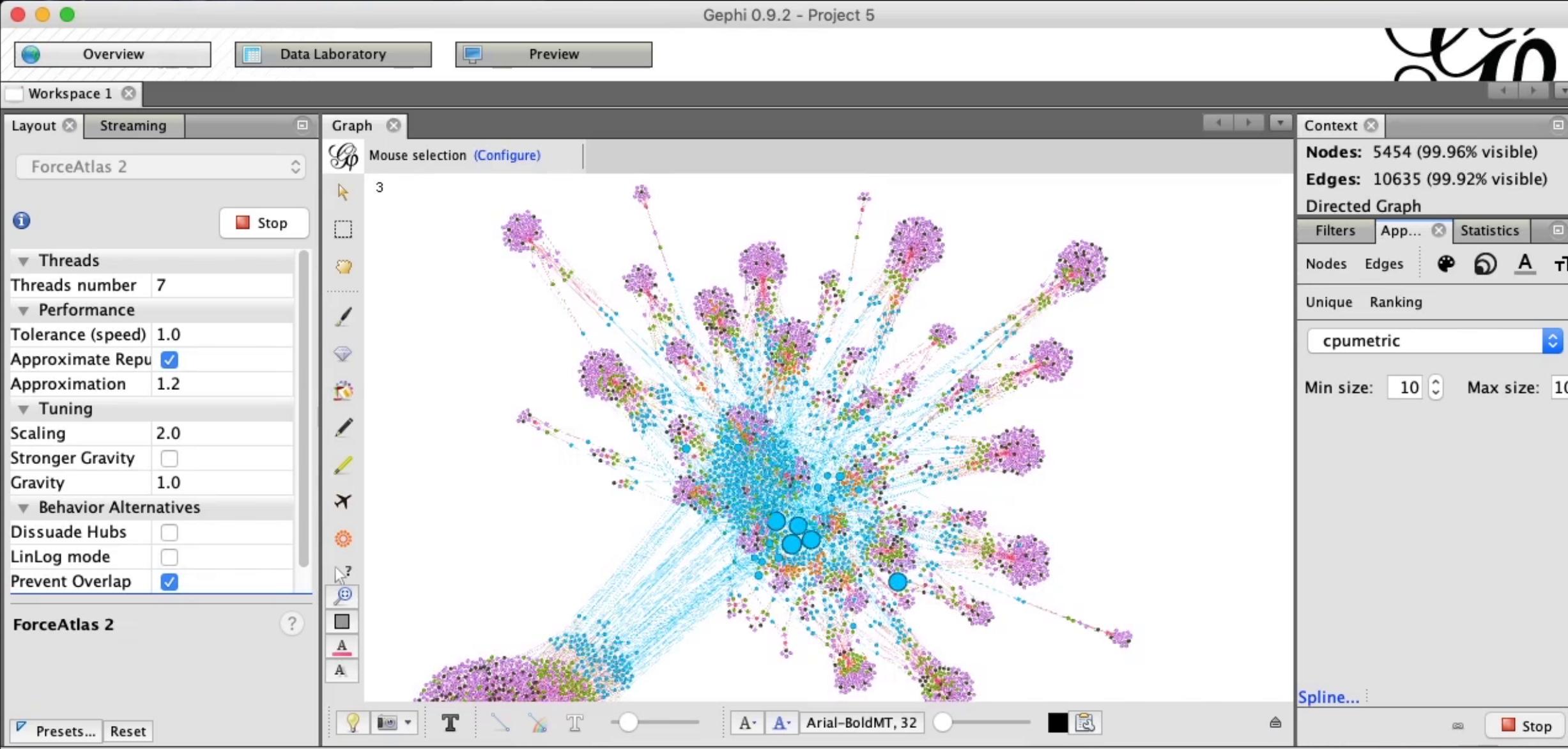
WATCH

CALL STACK

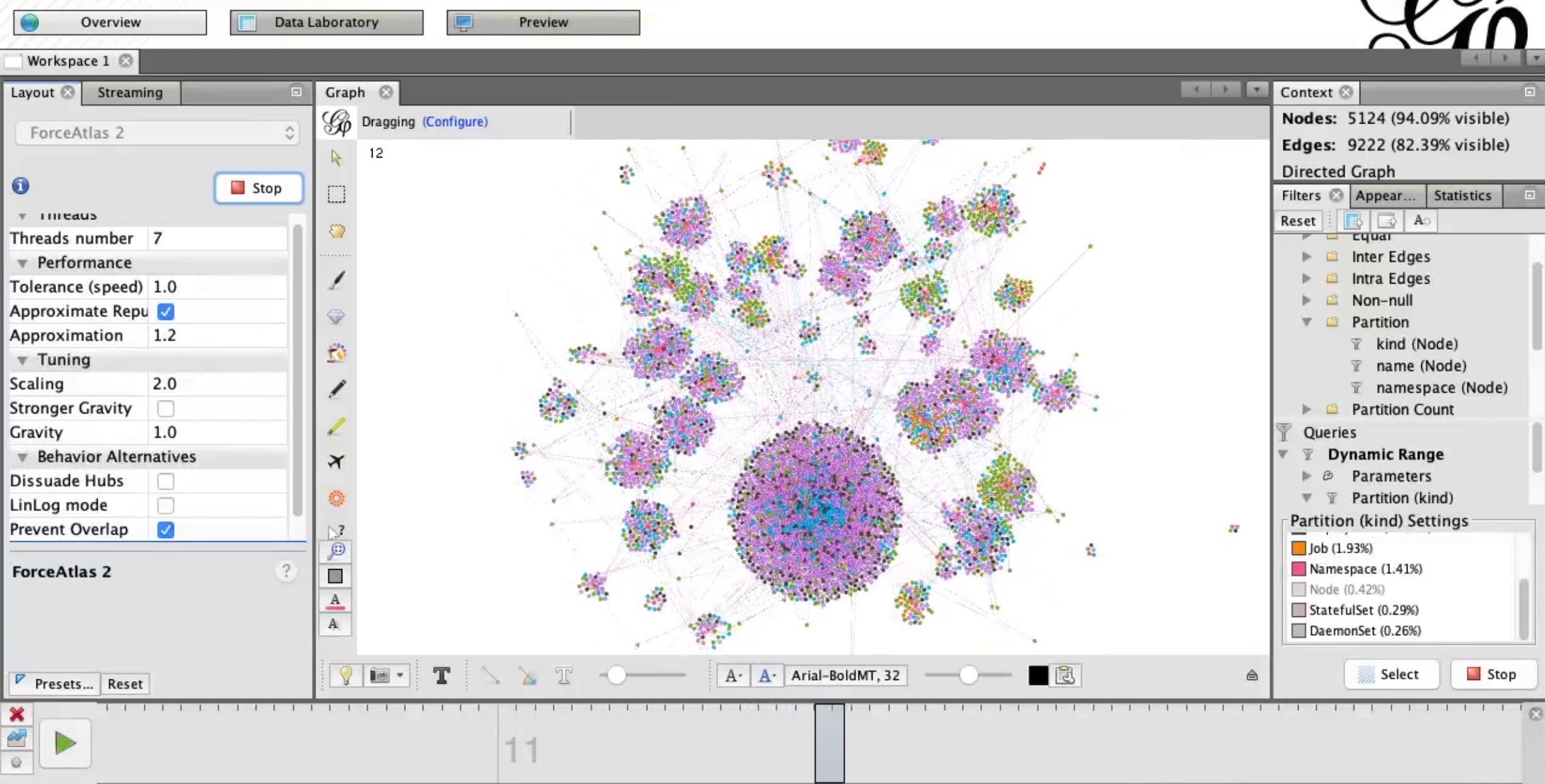
BREAKPOINTS

	Date	Time	Action	Object	Details
29378	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "a8be14c6-0b4c-4fcf-89d2-38ff0f0eb949", "Name": "thanos"}	
29379	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "df1d7532-3e47-46d9-98b5-3ad73bb9afa5", "Name": "analyt"}	
29380	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "c7ac17fe-b514-44c8-9230-d691713b0a75", "Name": "promet"}	
29381	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "edf4d706-77d1-4ce1-bd38-c223b9927c25", "Name": "grp49"}	
29382	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "69bf57f3-a411-469e-a692-b9fd127fae66", "Name": "grp45"}	
29383	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "5030e3ea-6c82-4046-9fc1-379571f0e538", "Name": "grp172"}	
29384	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "2809d900-118b-4528-be75-040b755b7601", "Name": "grp167"}	
29385	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "ebe7b1ea-6f75-4398-91ae-1ee413391f9d", "Name": "grp45"}	
29386	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "8f5ac127-dee2-459c-85fe-72dd1b08410a", "Name": "dev-fe"}	
29387	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "91820a80-9412-4a0e-8431-45019acf4f90", "Name": "grp49"}	
29388	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "4c98647f-5c85-447e-8b5d-b6e82b5a966a", "Name": "grp130"}	
29389	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "816a4245-b8b7-4f56-916a-882b6ebaa940", "Name": "grp76"}	
29390	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "a4fd81b6-5e4b-4995-b283-5acbf8ce2097", "Name": "grp130"}	
29391	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "e4cb2df1-d1fc-4745-9176-7c6aedc5cf6", "Name": "grp45"}	
29392	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "ec8991b5-3621-4bed-ad9f-2e9f0aec28d4", "Name": "fluent"}	
29393	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "6f2f4c46-5ad5-40a9-a35e-9bb4fe130a47", "Name": "monito"}	
29394	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "6f1adfbd-eb4f-4b29-ba23-fd45996d5d61", "Name": "grp100"}	
29395	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "88803912-a091-456d-aba8-a245c058820b", "Name": "grp10"}	
29396	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "5aac4901-e386-4bbb-9688-d4d007501ale", "Name": "kube-c"}	
29397	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "abf38692-0e71-44af-b011-fa2cd1b06867", "Name": "kube-p"}	
29398	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "9c0b707f-d30e-4c92-a887-4681c9c2f31d", "Name": "dev-gr"}	
29399	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "1fed4df3-ea9a-4bc7-9d01-8447fc2805b2", "Name": "grp51"}	
29400	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "4093cc19-4de3-489c-aebb-8d2a2187add8", "Name": "dev-gr"}	
29401	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "6b839832-b52d-4fb5-9b53-33ca30242f58", "Name": "grp51"}	
29402	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "1e8e02dc-8953-48a4-b2da-1f57a7d5fc5d", "Name": "fluent"}	
29403	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "82b42c2a-b0d7-4976-8cc1-5c67b9f02eb0", "Name": "grp76"}	
29404	2020/03/22	16:06:05	UPDATE,Pod,	{"Uid": "0cec49a8-71a4-4e4f-9d02-043647e82b3b", "Name": "dev-gr"}	

0 ▲ 24 Go Modules Ln 1, Col 1 Spaces: 4 UTF-8 LF Log







Overview Data Laboratory Preview

Workspace 1

Layout Streaming Graph Context

Nodes: 896 (16.45% visible)  
Edges: 868 (7.75% visible)  
Directed Graph

Filters Appear... Statistics

Reset Equal  
► Inter Edges  
► Intra Edges  
► Non-null  
▼ Partition  
    Y kind (Node)  
    Y name (Node)  
    Y namespace (Node)  
► Partition Count

Queries  
▼ Dynamic Range  
    Parameters  
    ▼ Partition (kind)

Partition (kind) Settings  
Service (15.08%)  
Deployment (11.59%)  
Job (1.93%)  
Namespace (1.41%)  
Node (0.42%)

Select Stop

ForceAtlas 2

Dragging (Configure) 31

Threads number 7

Performance

Tolerance (speed) 1.0

Approximate Repu

Approximation 1.2

Tuning

Scaling 2.0

Stronger Gravity

Gravity 1.0

Behavior Alternatives

Dissuade Hubs

LinLog mode

Prevent Overlap

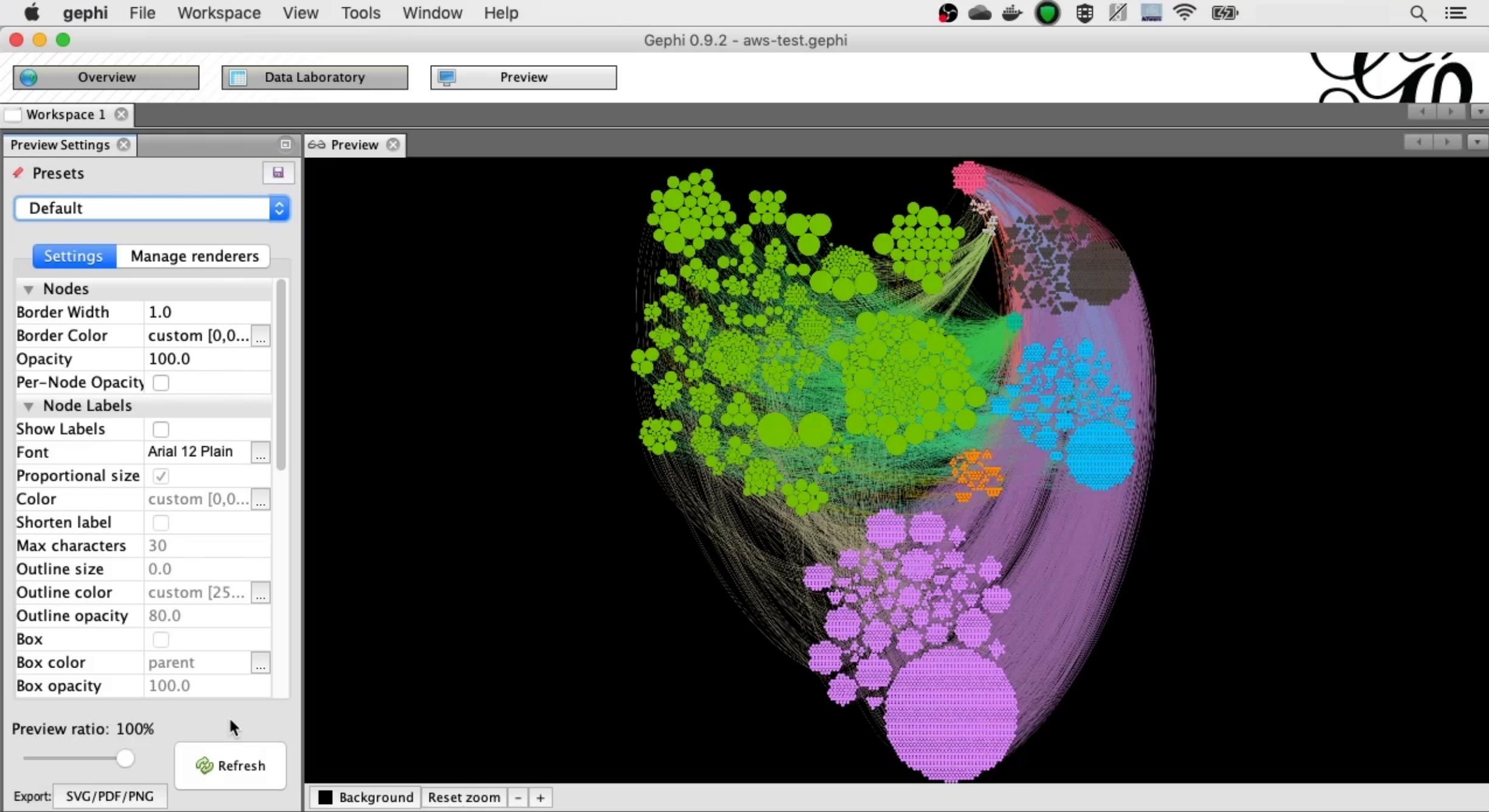
ForceAtlas 2

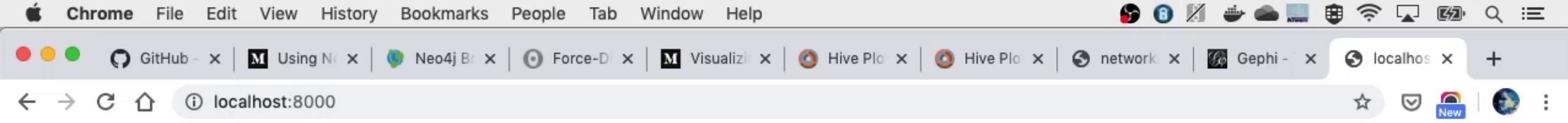
Presets... Reset

Arial-BoldMT, 32

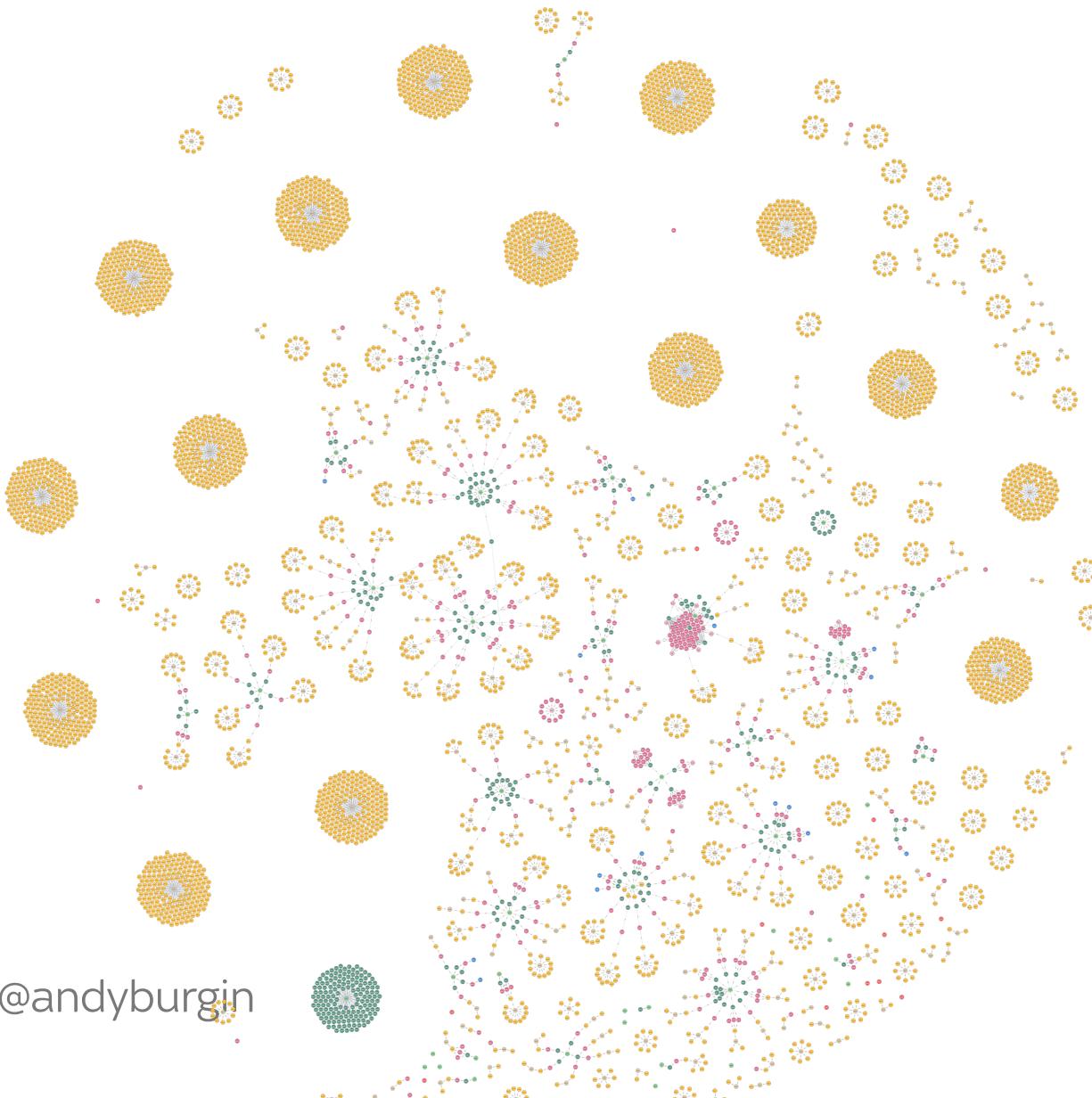
11

ForceAtlas 2



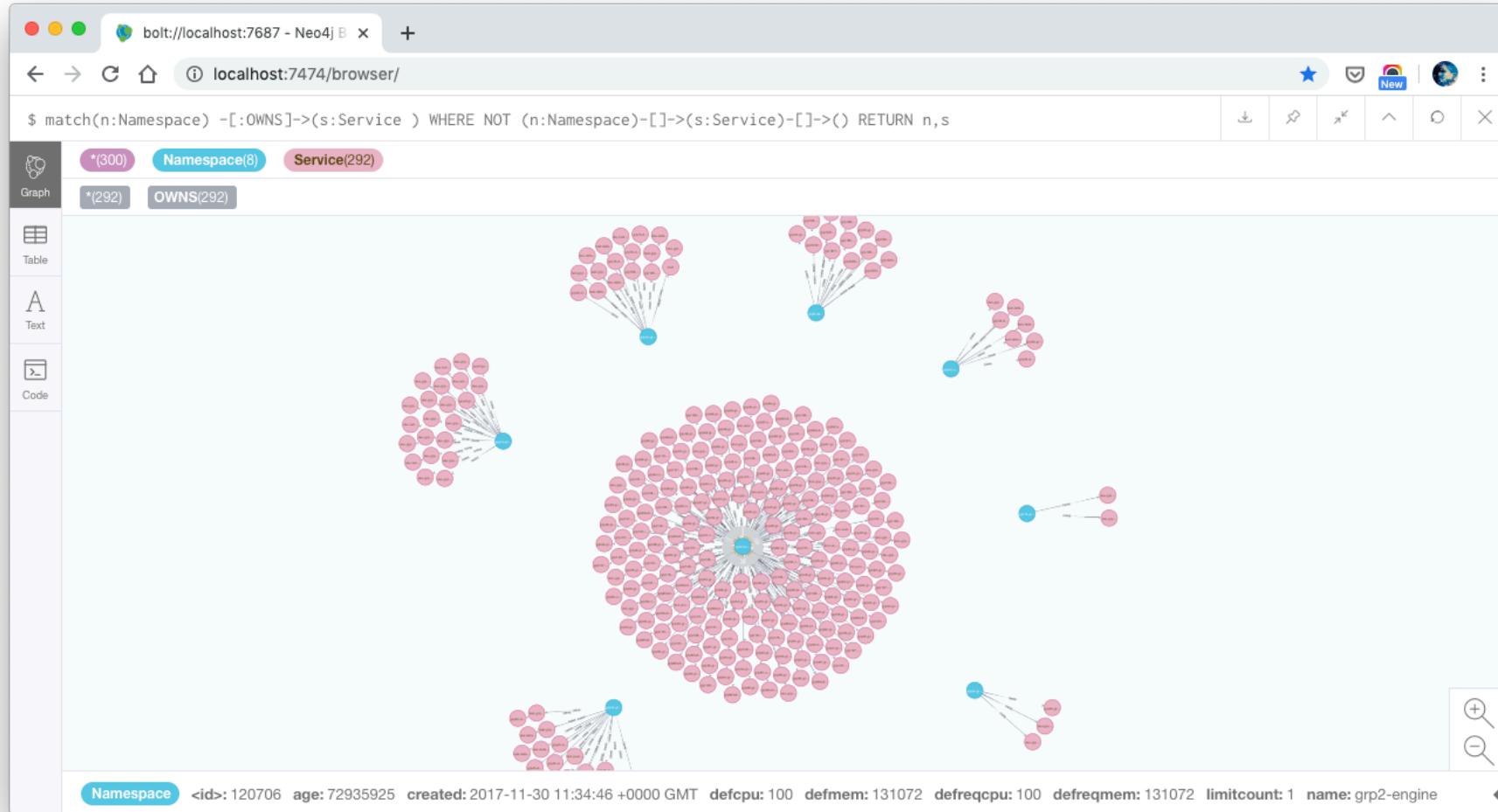


# Learnings

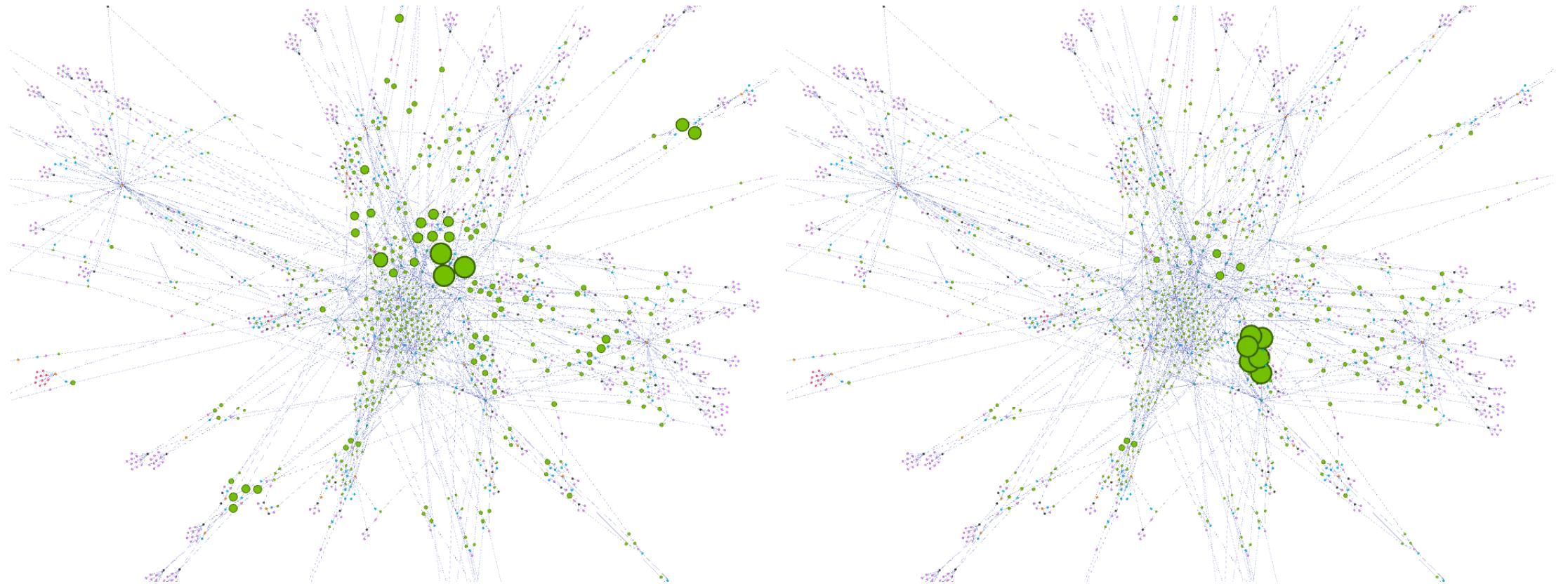


 @andyburgin

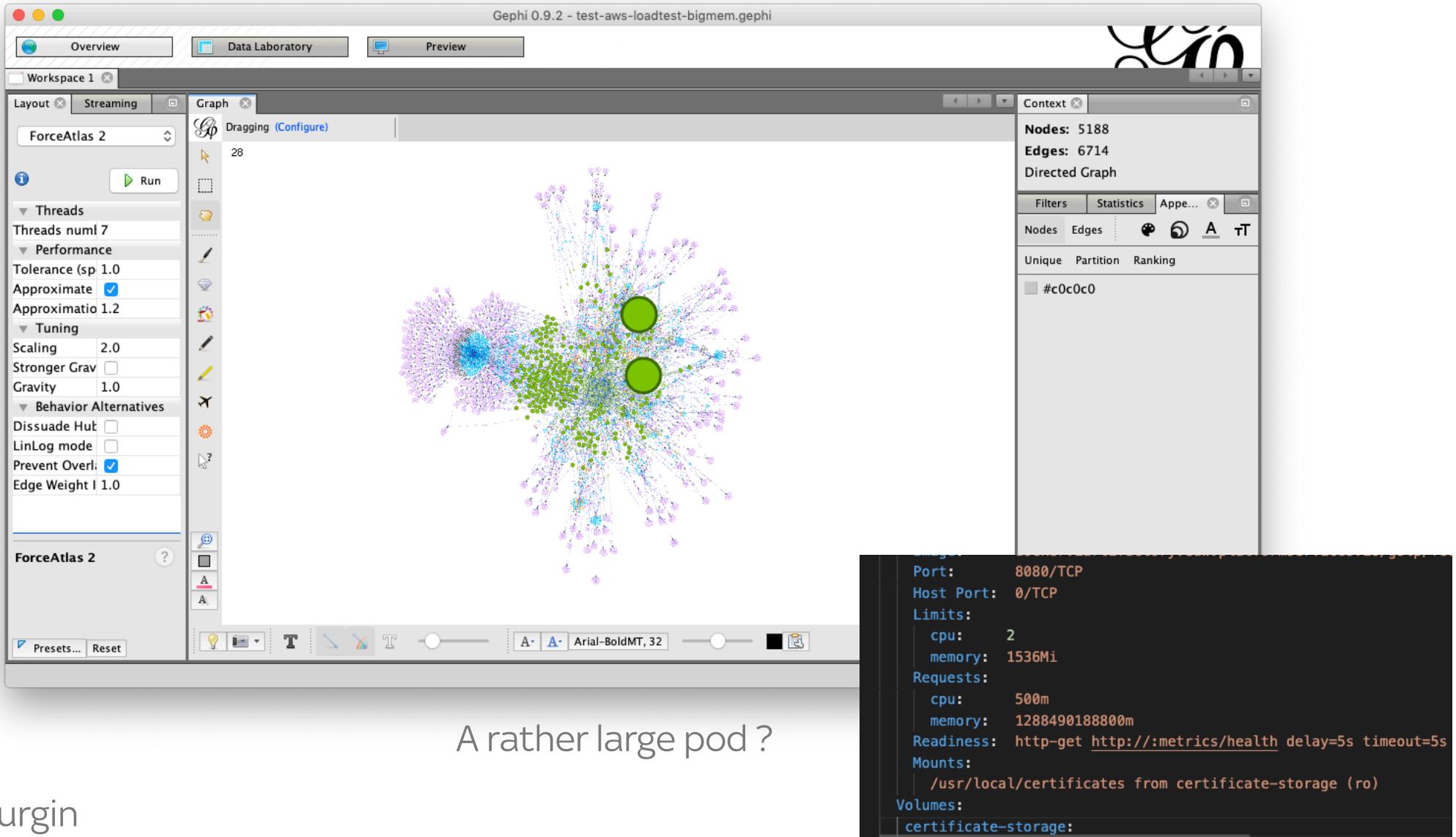
```
spec:  
  progressDeadlineSeconds: 2147483647  
  replicas: 1  
  revisionHistoryLimit: 2147483647  
  selector:  
    matchLabels:  
      app: test-test  
  strategy:  
    rollingUpdate:  
      maxSurge: 1  
      maxUnavailable: 1  
    type: RollingUpdate  
  template:  
    metadata:
```

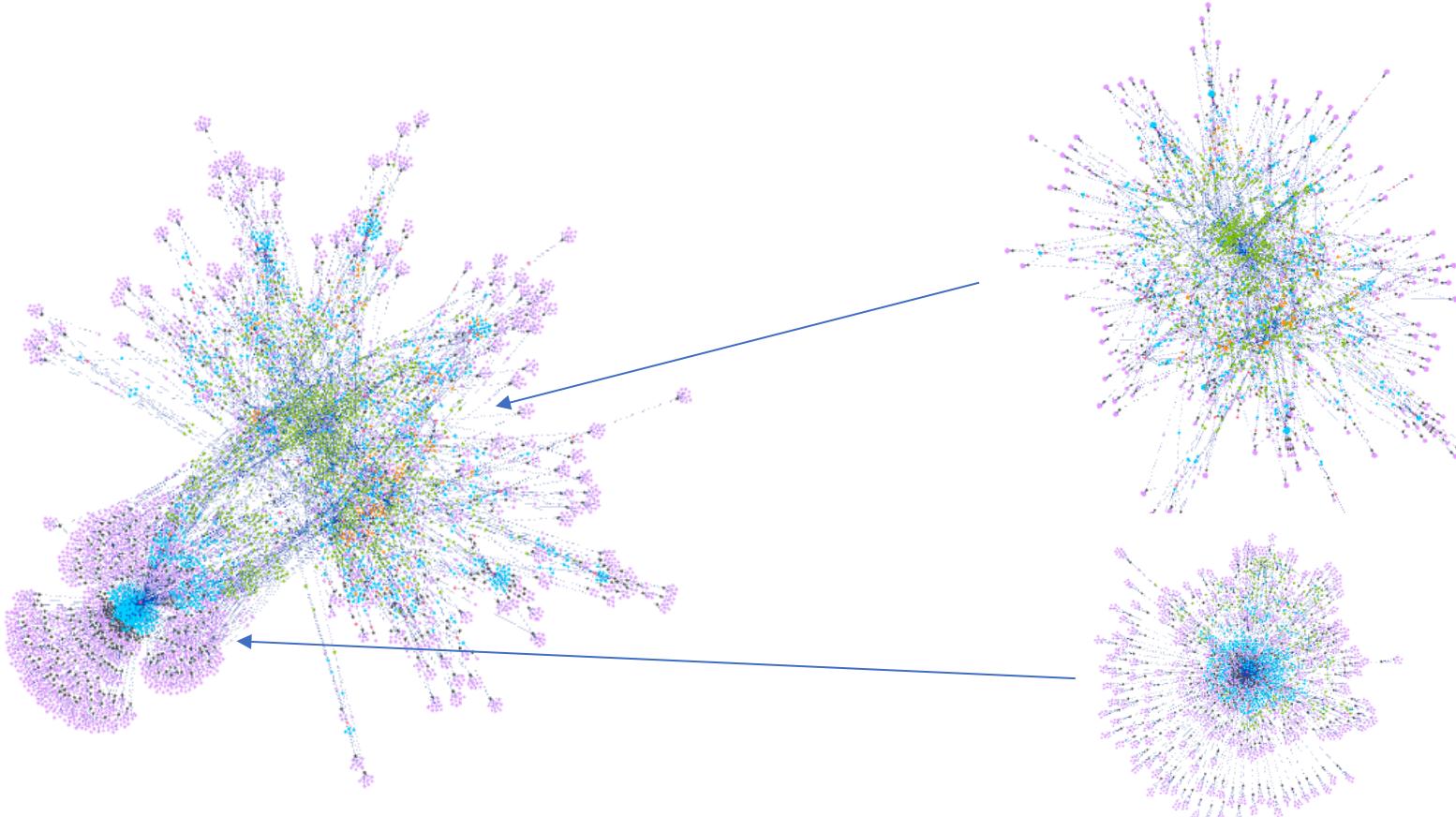


Simple Cypher query to find Services that don't expose any Pods

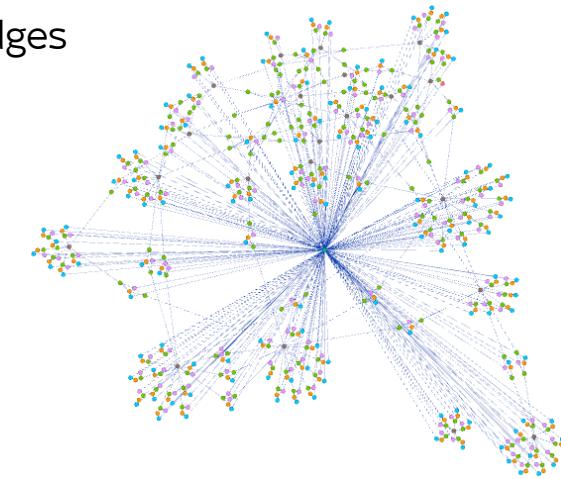
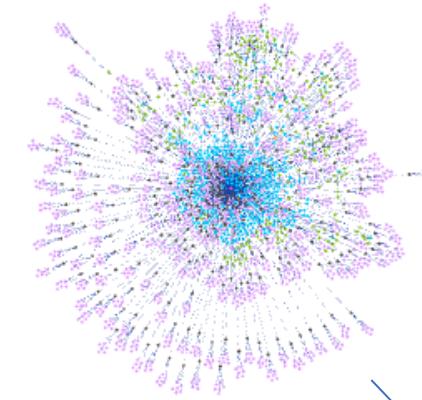
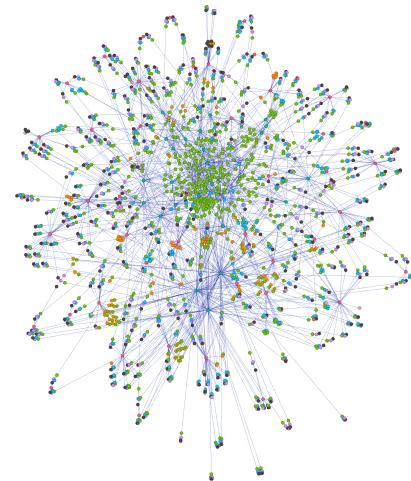
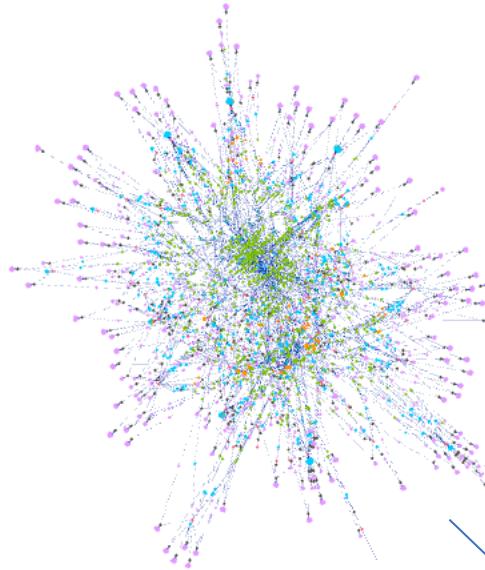


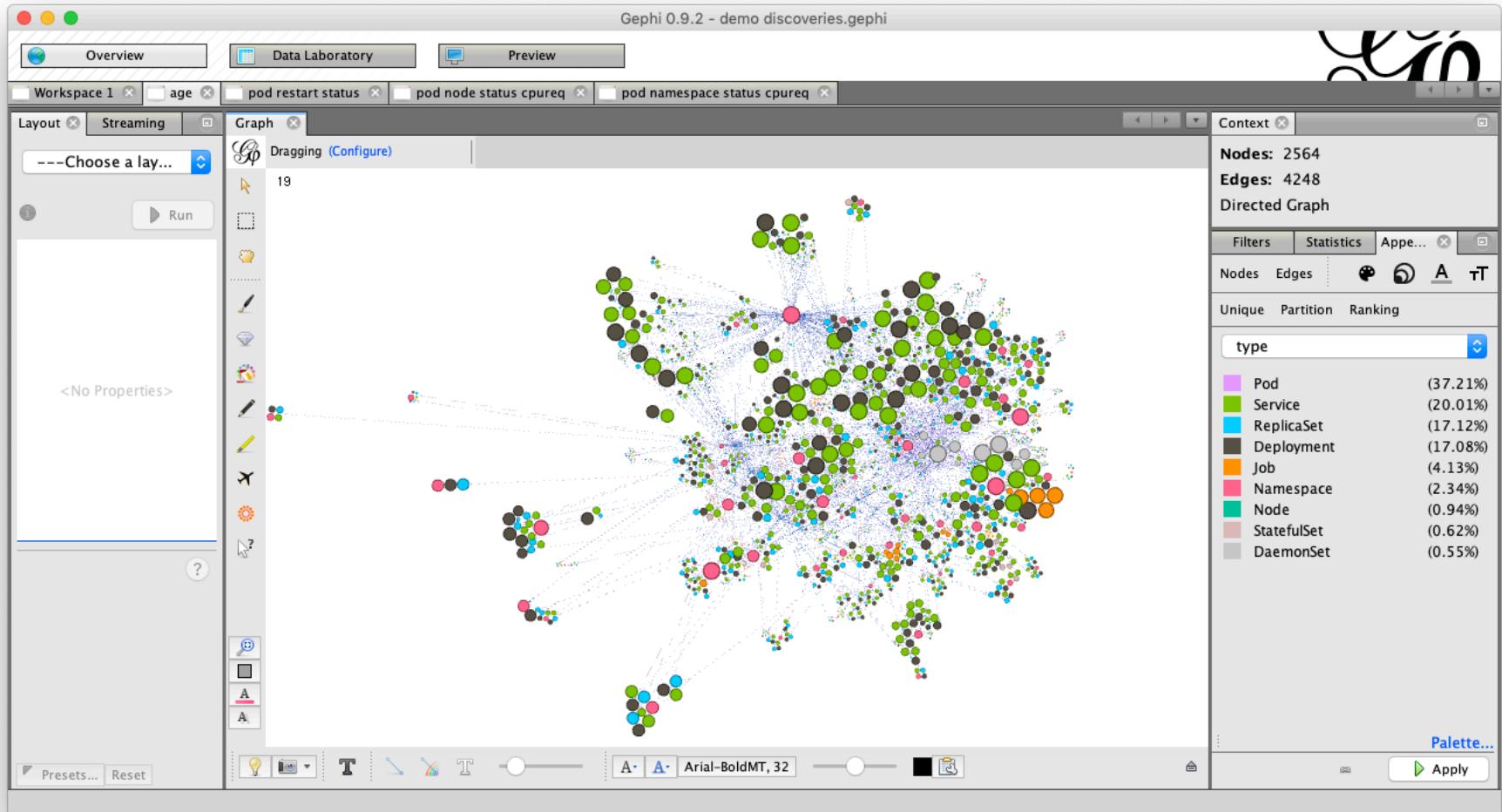
Pods sized by CPU and memory requests



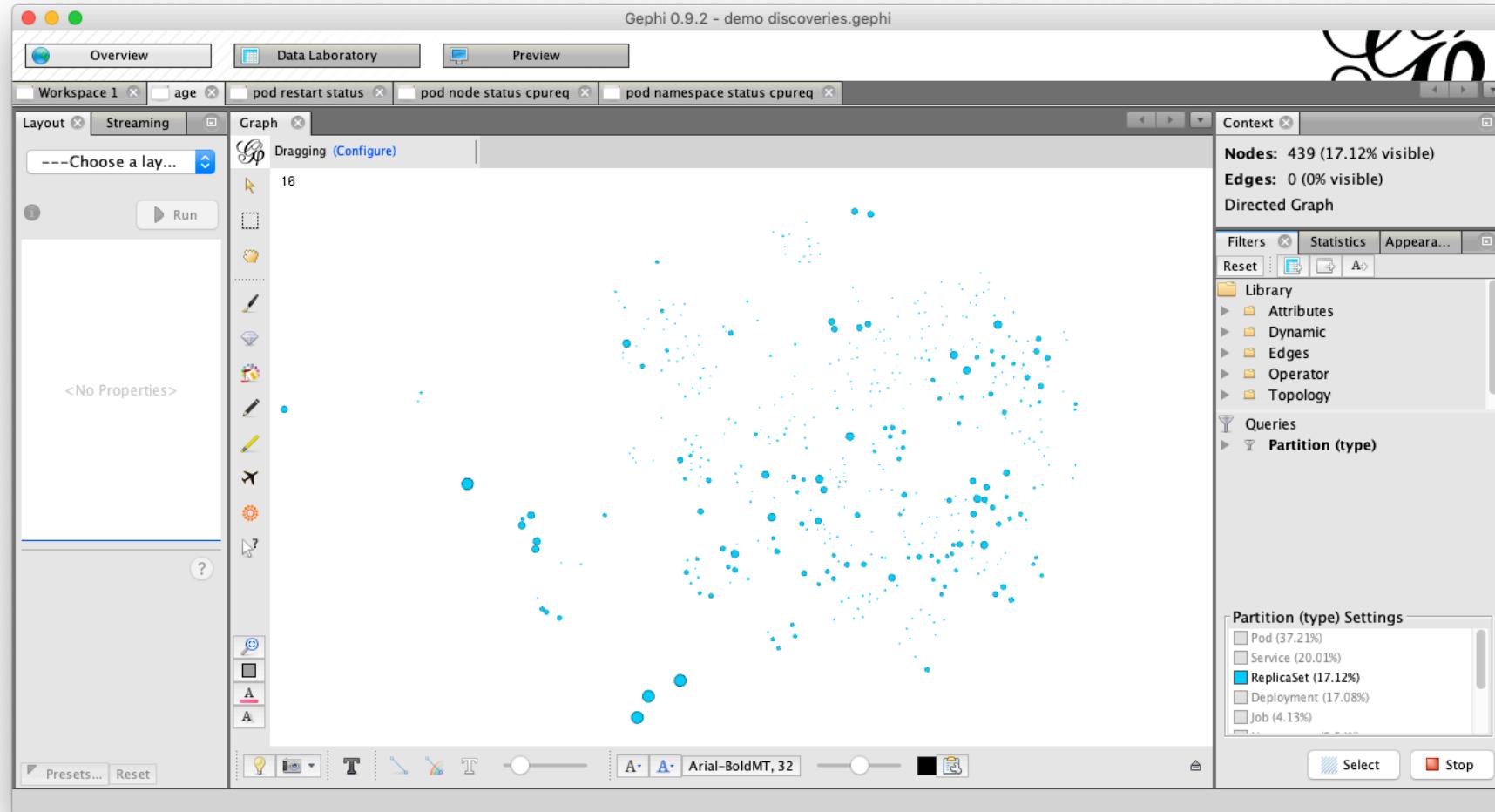


Splitting the hairball

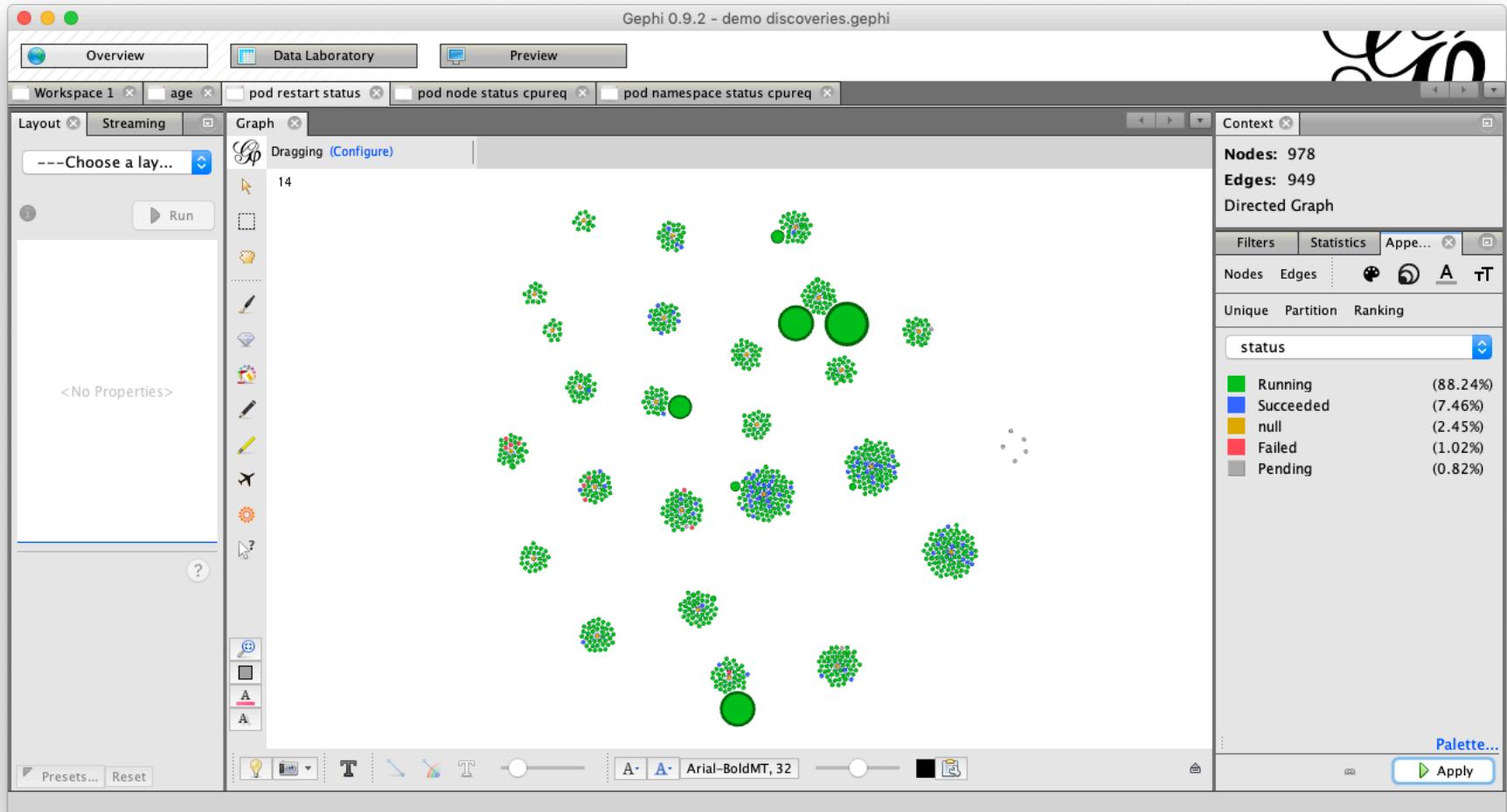




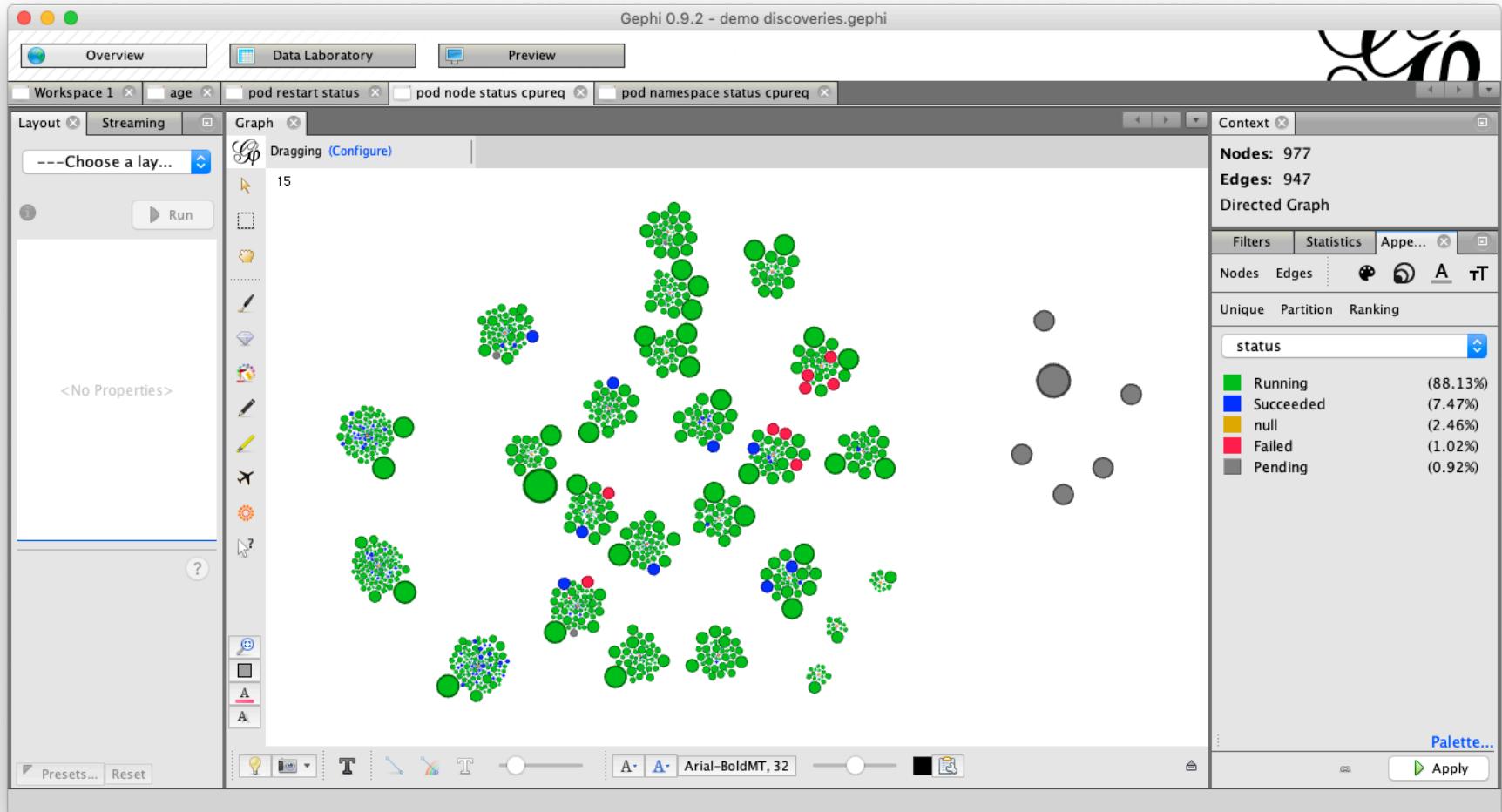
Simplified view of the cluster



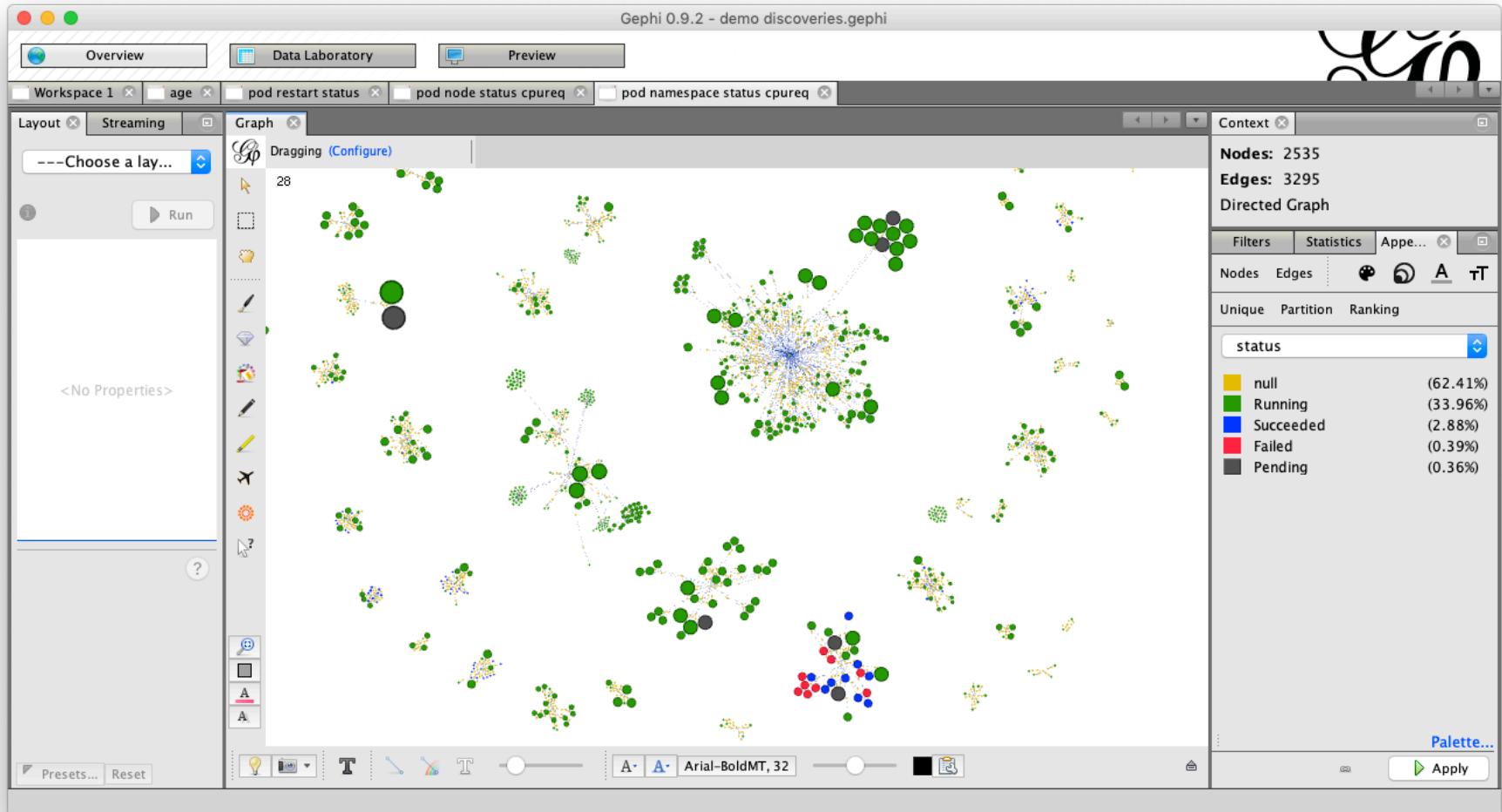
View just the active Replicasets, sized by age, rebuild and redeploy ?



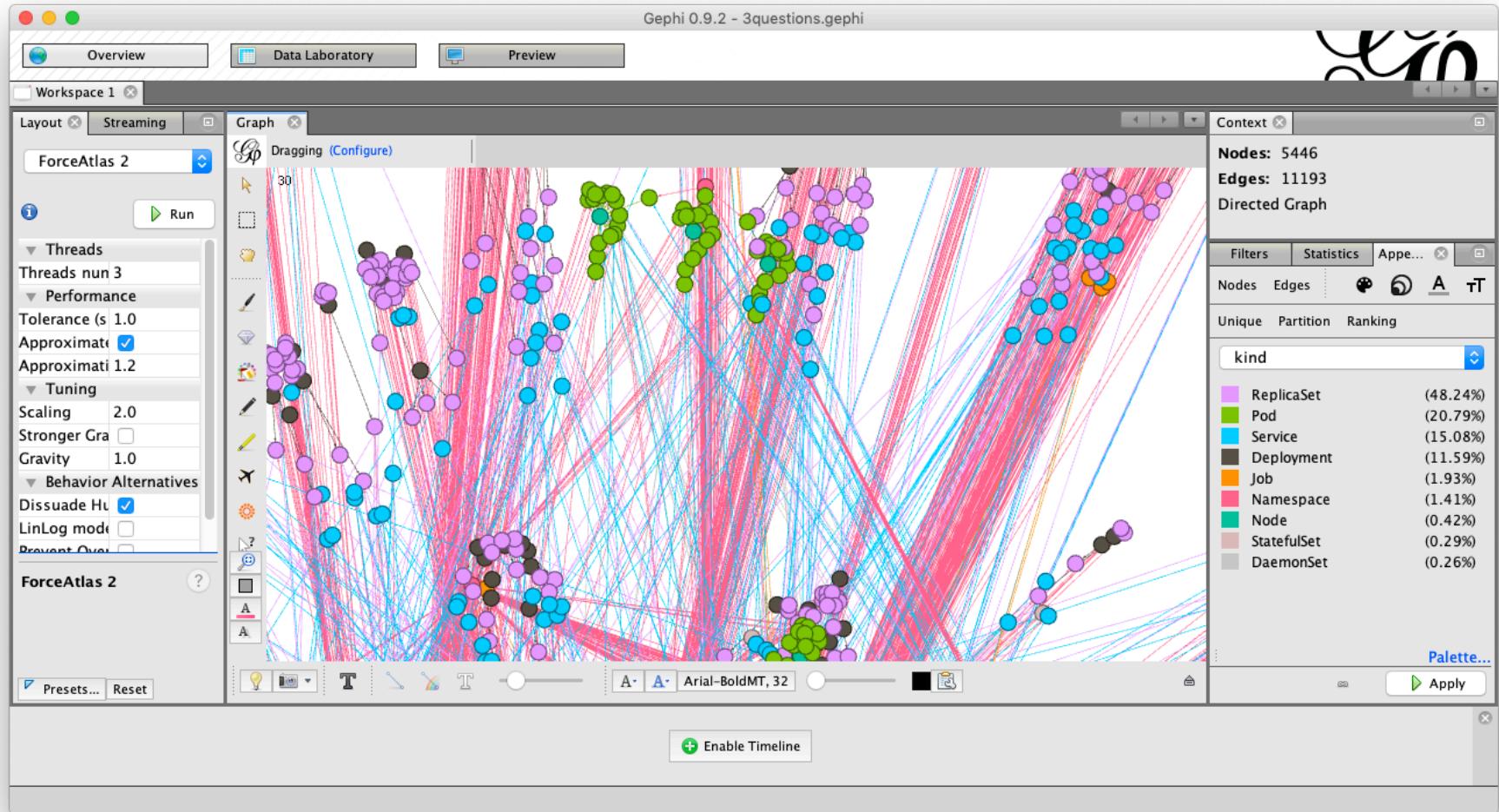
Pods sized by restarts, coloured by status



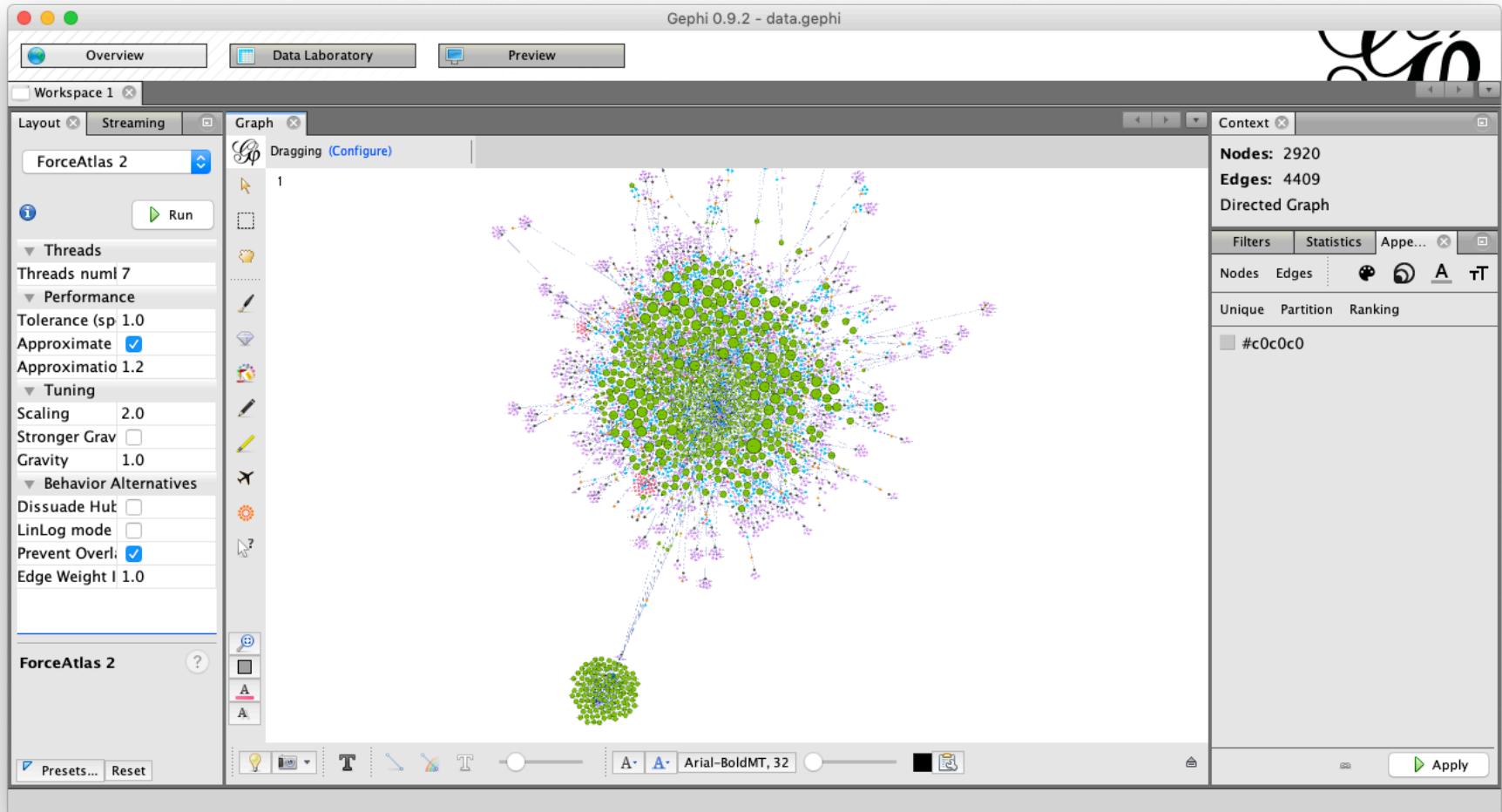
Pods sized by CPU request, coloured by status



Pods grouped by namespace, coloured by status and sized by CPU request.



3 question marks ? What does that mean ?



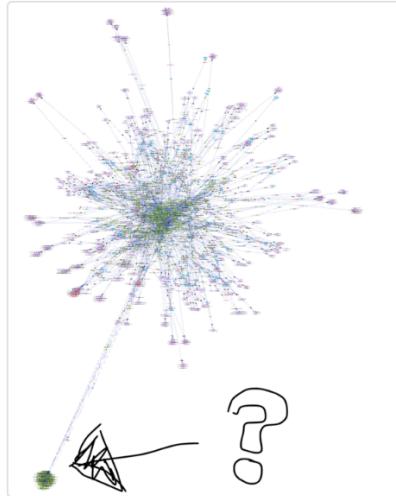
I reacted to an anomaly



**Andrew Burgin** 15:36

So when you do a bit of L&D and find something odd on your test cluster

Screenshot 2019-10-11 at 15.33.38.png ▾



anyone know anything about sysbeanch ? (edited)



15:36

It's what [REDACTED] is using to perf test the physicals

YES 1



5:38

Yesterday I ran a single pod with as much memory as I could, I got up to nearly 450GB before the worker fell over.



**Andrew Burgin** 15:39

ah yes worker tst31 (edited)

phew!

Whoops!

# What's Next ?

# What's Next ?

Kubernetes Objects:

- Explore Nodes objects.
- More real-time data (extracted from Metrics server and Prometheus)
- Analyse configmaps and secrets usage.
- Volumes and storage usage.
- RBAC and permissions.

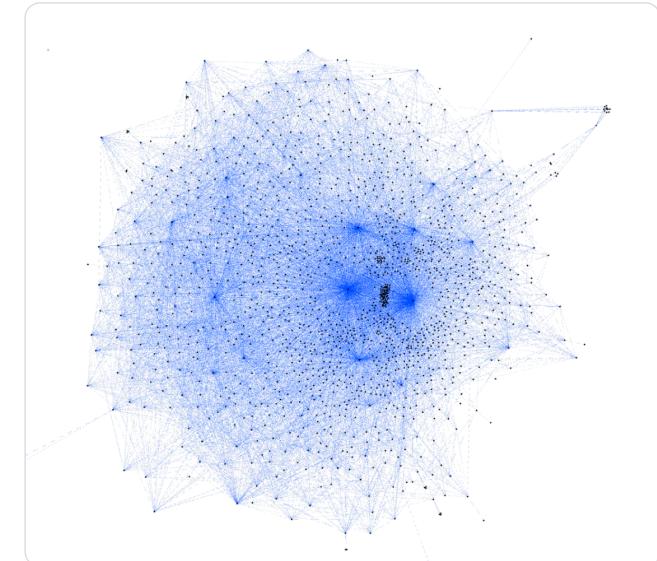
# What's Next ?

Networks:

- Network Policies
- Services
- Service Mesh



1500 microservices at @monzo; every line is an enforced network rule allowing traffic



7:47 PM · Nov 1, 2019 · [Twitter Web App](#)

639 Retweets 2.6K Likes

# What's Next ?

Networks:

- Network Policies
- Services
- Service Mesh

<input checked="" type="checkbox"/> <b>Network Overview</b>		
Average Degree	<input type="button" value="Run"/>	<input checked="" type="radio"/>
Avg. Weighted Degree	<input type="button" value="Run"/>	<input checked="" type="radio"/>
Network Diameter	<input type="button" value="Run"/>	<input checked="" type="radio"/>
Graph Density	<input type="button" value="Run"/>	<input checked="" type="radio"/>
HITS	<input type="button" value="Run"/>	<input checked="" type="radio"/>
Modularity	<input type="button" value="Run"/>	<input checked="" type="radio"/>
PageRank	<input type="button" value="Run"/>	<input checked="" type="radio"/>
Connected Components	<input type="button" value="Run"/>	<input checked="" type="radio"/>
<input checked="" type="checkbox"/> <b>Node Overview</b>		
Avg. Clustering Coefficient	<input type="button" value="Run"/>	<input checked="" type="radio"/>
Eigenvector Centrality	<input type="button" value="Run"/>	<input checked="" type="radio"/>
<input checked="" type="checkbox"/> <b>Edge Overview</b>		
Avg. Path Length	<input type="button" value="Run"/>	<input checked="" type="radio"/>
<input checked="" type="checkbox"/> <b>Dynamic</b>		
# Nodes	<input type="button" value="Run"/>	<input checked="" type="radio"/>
# Edges	<input type="button" value="Run"/>	<input checked="" type="radio"/>
Degree	<input type="button" value="Run"/>	<input checked="" type="radio"/>
Clustering Coefficient	<input type="button" value="Run"/>	<input checked="" type="radio"/>

The image is a composite of three parts. On the left is a network graph visualization titled "DETECTION" showing nodes in yellow, pink, and purple clusters connected by red edges. In the center is a large red logo resembling a stylized letter 'C'. On the right is a photograph of a woman with short blonde hair, wearing glasses and a patterned blouse, standing on a stage and speaking to an audience.

DETECTION

@techiwatt

SUBSCRIBE;

<https://www.youtube.com/watch?v=0G5O1ffYIPI>

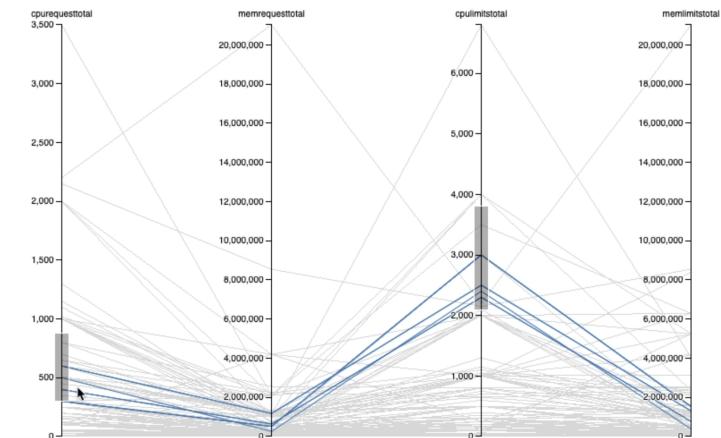
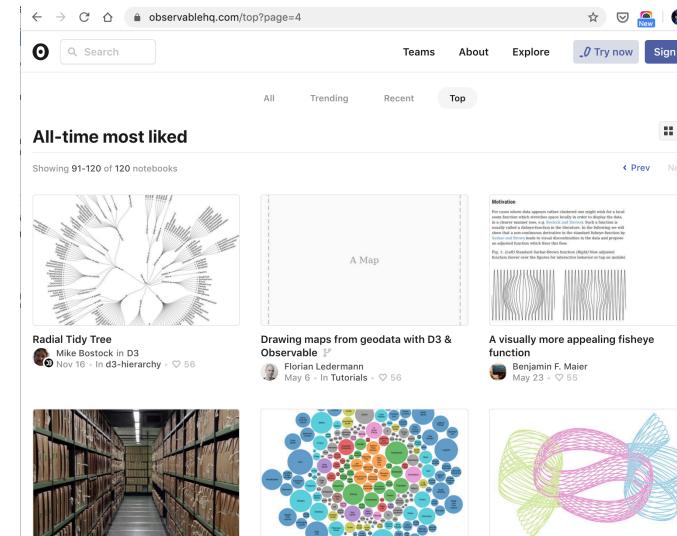
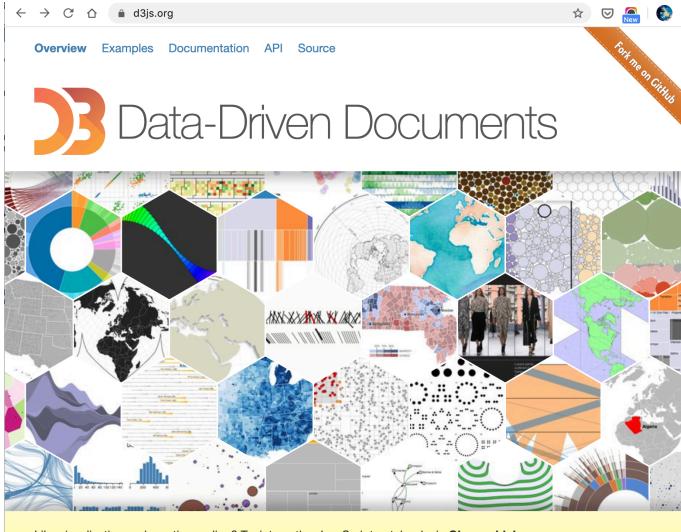
# Gephi – The Future

- Single Node and Edge type.
- The future in it's current form is in question...  
<https://gephi.wordpress.com/2018/11/01/is-gephi-obsolete-situation-and-perspectives/>  
<https://gephi.wordpress.com/2019/02/02/exploring-the-dystopian-future-of-a-javascript-gephi/>
- Can be a little quirky!
- How to get started links in the resources.



Eduardo Ramos Ibáñez, Mathieu Bastian  
and Mathieu Jacomy

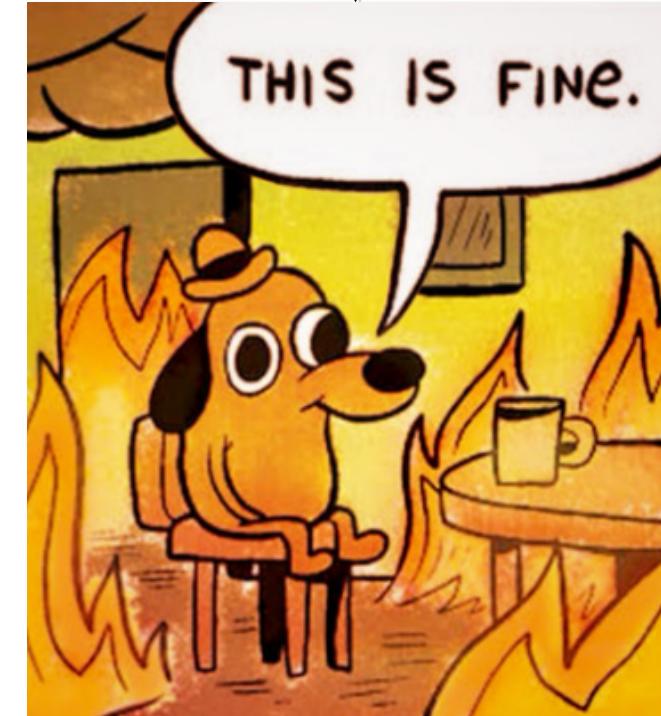
# Other Visualisations -D3



# DevOps FTW



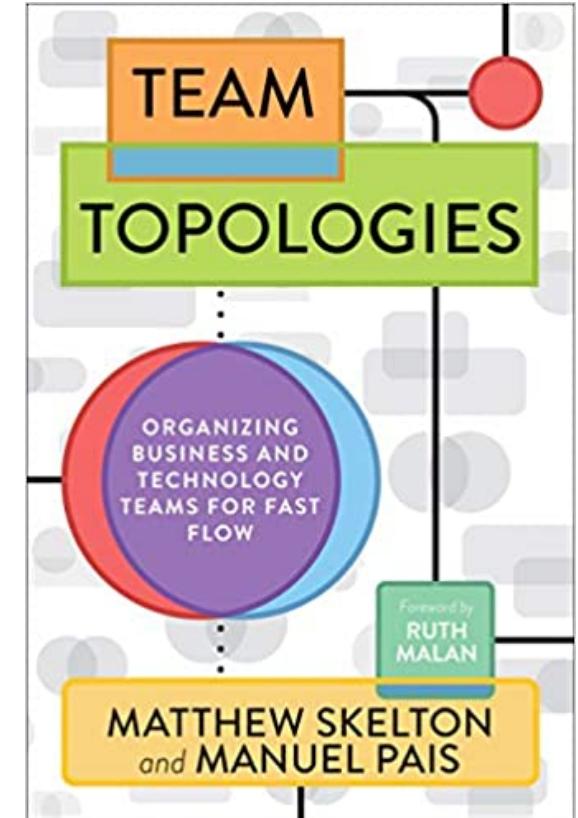
Handoffs →



<- Feedback

# Collaboration

- Steam-aligned team.
- **Enabling team.**
- Complicated Subsystem team.
- Platform team.



# Summary

# Stella Report ?

4 Supporting work through improved visualizations

**SN4U CATCHERS STELLA**  
Report from the SNAFUcatchers Workshop on Coping With Complexity  
Brooklyn NY, March 14-16, 2017

[Download PDF Version](#)

1. tl;dr and Executive Summary  
 1.1 tl;dr  
 1.2 Executive Summary

2. Introduction  
 2.1 About the SNAFUcatchers consortium and the STELLA meeting  
 2.2 The focus on handling anomalies  
 2.3 The above-the-line/below-the-line framework

3. Cases  
 3.1 Catching the Apache SNAFU  
 3.2 Catching the Travis CI SNAFU  
 3.3 Catching the Logstash SNAFU  
 3.4 Anomalies in the cases  
 3.4.1 Features of the anomalies  
 3.4.2 Features of the anomaly responses  
 3.4.3 Supporting communication and uncertainty  
 The role of search  
 Evolutionary system representations  
 Generating hypotheses  
 Basic tools  
 Coordination  
 Communications in joint activity  
 Shared artifacts  
 The consequences of escalating consequences  
 Managing risk  
 Goal setting  
 3.5 Observations on the postmortem process

4. Themes  
 4.1 Capturing the value of anomalies  
 4.1.1 Technical issues in postmortems  
 4.1.2 Social issues in postmortems  
 4.2 The paradox of sanction in the aftermath of anomalies  
 4.3 Controlling the costs of coordination during complex responses  
 4.3.1 Offloading work to low-tempo periods  
 4.3.2 Providing expertise on demand  
 4.3.3 Supporting communication and coordination with tools  
 4.4 Supporting anomaly response through improved visualizations  
 4.4.1 Understanding collective work in context  
 4.4.2 A starting point  
 4.5 Strange loops dependencies  
 4.6 Debt  
 4.6.1 Technical debt  
 Origins of the debt metaphor  
 Technical debt refactoring  
 Technical debt 25 years on  
 4.6.2 Dark debt  
 5. Paths to progress for progress on coping with complexity  
 6. Back matter  
 6.1 Acknowledgment  
 6.2 Acknowledgements  
 6.3 Suggested citation for this report  
 7. References

**Winter storm STELLA**



Woods' Theorem: As the complexity of a system increases, the accuracy of any single agent's own model of that system decreases rapidly.

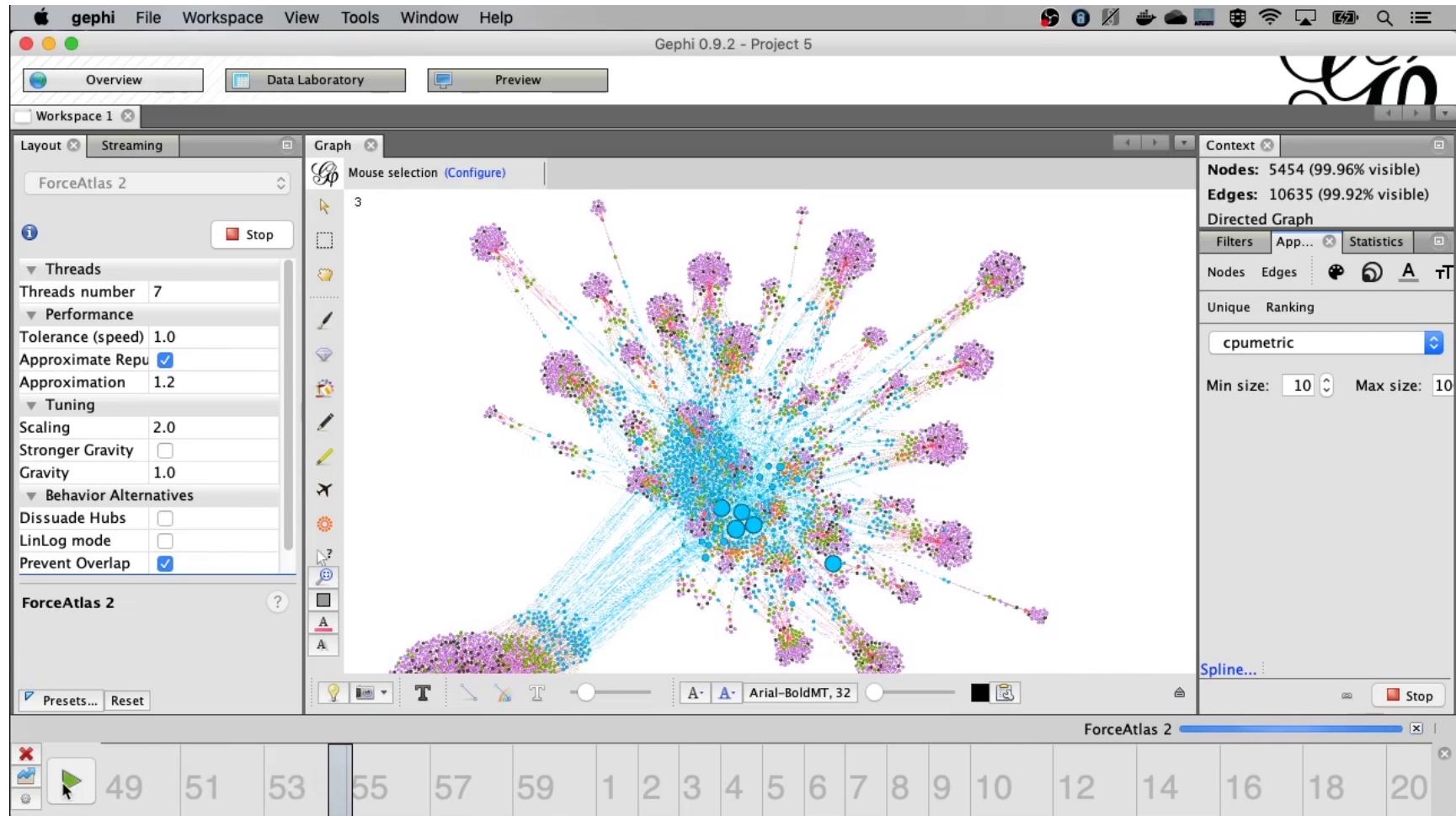
**1. tl;dr and Executive Summary**

A comprehensive workshop of high end techs reviewed postmortems to better understand how engineers cope with the complexity of anomalies (SNAFU and SNAFU catching episodes) and how to support them. These cases reveal common themes regarding factors that produce resilient performances. The themes that emerge also highlight opportunities to move forward.

**1.2 Executive Summary**

Current generation internet-facing technology platforms are complex and prone to brittle failure. Without the continuous effort of engineers to keep them running they would stop working -- many in days, most in weeks, all within a year. These platforms remain alive and functioning because workers are able to detect anomalies, diagnose their sources, remediate their effect, and repair their flaws and do so ceaselessly -- SNAFU Catching. Yet we know little about how they accomplish this vital work and...

<http://stella.report>



Point in Time Analysis

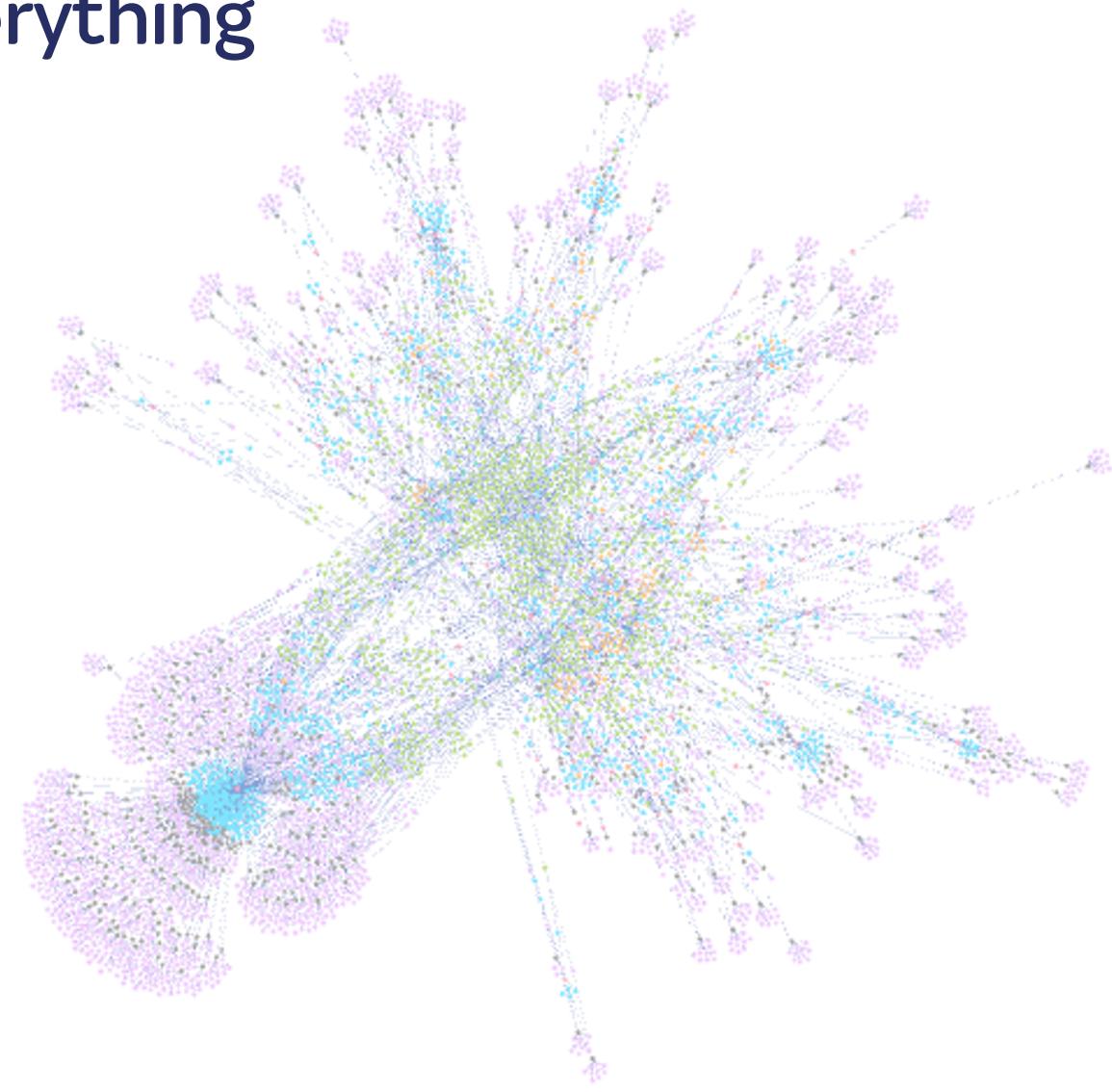
# Context and Perspective is Everything

It's hard to understand the hairball.

Remove as much noise as you can.

Select a perspective and apply the context

- Workload.
- Compute.
- Connections.
- Security.



# Just One More Thing....

# Resources

# Resources

How Your Systems Keep Running Day After Day - John Allspaw

<https://www.youtube.com/watch?v=xA5U85LSk0M>

<http://stella.report>

Kvizz

<https://github.com/afbjorklund/kvizz>

[thread/SBDE64DAIVTLIYRGVMESUPR5STRRFCGW/](https://github.com/afbjorklund/kvizz/thread/SBDE64DAIVTLIYRGVMESUPR5STRRFCGW/)

Weavescope

<https://www.weave.works/oss/scope/>

# Resources

Cockpit Kubernetes plugin (removed)

<https://lists.fedorahosted.org/archives/list/cockpit-devel@lists.fedorahosted.org/thread/SBDE64DAIVTLIYRGVMESUPR5STRRFCGW/>

Using Neo4J to visualize a Kubernetes cluster – Bajal

<https://medium.com/@bajalm/using-neo4j-to-visualize-a-kubernetes-cluster-1d2f5190eb93>

Force-Directed Graph - Mike Bostock

<https://observablehq.com/@d3/force-directed-graph>

## Resources

Visualizing Graphs in 3D with WebGL - Michael Hunger

<https://medium.com/neo4j/visualizing-graphs-in-3d-with-webgl-9adaaff6fe43>

Hive Plots - Rational Network Visualization – Farewell To Hairballs - Martin Krzywinski

<http://www.hiveplot.com/>

Gephi - The Open Graph Viz Platform

<https://gephi.org/>

Introduction to GEPHI - University of Kentucky Libraries

<https://www.youtube.com/watch?v=2FqM4gKeNO4>

# Resources

Gephi Tutorials - Jen Golbeck

[https://www.youtube.com/playlist?list=PLk\\_jmmkw5S2BqnYBqF2VNPsY93-ze49](https://www.youtube.com/playlist?list=PLk_jmmkw5S2BqnYBqF2VNPsY93-ze49)

Gephi Layout tutorial

<https://www.slideshare.net/gephi/gephi-tutorial-layouts>

Social Network Analysis - Lada Adamic - University of Michigan

<http://www-personal.umich.edu/~ladamic/courses/>

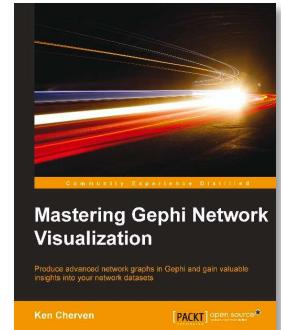
[https://www.youtube.com/playlist?list=PL2rR6Wa-StjYOW7v6J8\\_npck6EDOKEbCN](https://www.youtube.com/playlist?list=PL2rR6Wa-StjYOW7v6J8_npck6EDOKEbCN)

# Resources

Mastering Gephi Network Visualization

Ken Cherven

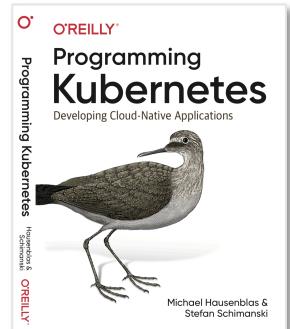
<https://www.packtpub.com/gb/networking-and-servers/mastering-gephi-network-visualization>



Programming Kubernetes - Developing Cloud Native Applications

Michael Hausenblas and Stefan Schimanski

<https://programming-kubernetes.info/>



Go Source for extracting object data from k8s and writing log/gexf and Neo4j

<https://sbg.technology/2020/04/28/vis-complex-systems/>

# Resources

D3 Data Driven Documents

<https://d3js.org/>

Observable

<https://observablehq.com/explore>

Parallel Coordinates - Jason Davies

<https://bl.ocks.org/jasondavies/1341281>

GOTO 2019 • Explore Microservices Architecture with Graph Theory & Network Science • Nicki Watt

<https://www.youtube.com/watch?v=0G5O1ffYIPI>

# Thank You