UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science

COS 301 - Software Engineering

Amazon Network Visualizer Demo 3

Not-Like-This

Authors:	*Student number:*
Jedd Schneier	u13133064
Daniel King	u13307607
Muller Potgieter	u12003672

September 11, 2016

Contents

1 Vision

The Network Visualizer is intended to be used by registered Amazon Web Services (AWS) users. The visualizer is primarily aimed at consumers of AWS, in order to provide a simple and clear representation of the various networks' structures.

In order to access the visualizer, the user must first submit their AWS password and secret password. The visualizer then attempts to access the server, using the provided passwords. If this is successful, the visualizer will scan the specified network and log the nodes (such as instances and VPC's) and their relationships. It uses this information to construct a tree-esque representation of the network.

Using this representation, it is then translated into HTML. Making use of the vis.js library, the structure of the network is presented in a clear, visual hierarchial structure. The page also allows the user to specify which region they wish to be scanned and represented.

2 Scope

- The user requires a valid Amazon Web Services account (or access to one), in order to make use of the visualizer.
- The user requires an internet capable device and a modern browser to access the visualizer's page.

3 Architectural Requirements

3.1 Architecture Scope

Layered Architecture: RESTFUl Architecture Threading Pool tactic:

3.2 Quality Requirements

3.2.1 Critical

1. Performance: The first and perhapse msot important critical requirment. The system is working with massive amounts of data. the system is redundant if it can not process and shwo that data quickly. The majority of architectual design ahs been don to improve perfroamnce. To keep it withib its 3 second throughput requiremnt.

2. Scalability: This is one of the major quality requirements for the service. As stated from the start of the project. The system will have to work with very large networks in future and thus the algorithms within will need to scale well such that they can work with these large networks and so that performance costs are minimized when these large networks are used. So far our Performance requirements matches our Scale.

3.2.2 Important

1. Security: From our own experince we know how important securoty is with this system.As we are working with sensitive client info (links to credit cards), it is imperative that we make sure the system is secure and does not allow for this iinformation to go to unwanted 3rd parties or be compromised in anyway.

2. Usability: Usability is very important for if the system is not easy, comfortable and pleasuarabel to use then the client may not feel a use in using it at all. However it is not important as perfrormance or Scaleabilty as at higher levels regardless of usability the ssytem will be required to meet thsoe requirments. The system is going to be used by the Amazon clients who will not necessarily have programming knowledge thus the end product needs to be catered to these users such that they are able to use the program efficiently.

3.2.3 Nice To Have

Maintainability: The system is going to be used by the Amazon clients who will not necessarily have programming knowledge thus the end product needs to be catered to these users such that they are able to use the program efficiently.

3.3 Architectural Constraint

The user must have an existing and valid account with the Amazon services. The user must be using a virtual network. the user has to havea valid credit card to set up their account. user msut havea relitavily stable preferbly uncapped internet connection. USer must also set up access key and Secrete key on their end.

3.4 Access and integration channels

The information regarding the networks is taken via the Amazon API services. - The user should login via the existing Amazon site and the visualizer should integrate into the site itself. - The Front end must intergrate through the REST server with the backend. -The scanner must intergrate with the Shared Buffer. -The Scanner mustt intergrate with AWS. -The interface must intergrate on all browsers and with standard web technologies 1.4 Architectural constraints

4 Architectural Design

Layered Architecture:

Our core architectural design follows a standard Client-Applciation-Server layered architecture. As seen below, the "Server" (not to be confused with the Application server) is the AWS server we connect to through their API. The product does not persist to only reads from it and is we have no controll over its functionality it will be left for an intergratio requirment and will not be disccues furhter in this section. Additional information on the AWS ssytem and their API can be found here https://aws.amazon.com/documentation/ . Previously we useda bridging layer ,ADAPTER, to join the Application and Server layer, but it has sesne been incorporated into the Scanner to reduce communciation overhead and facilitate the threading tactics. The main backend of the system lies on the Application layer, which is broken down into the follwoing components:

1. Scanner: Most important component, constructs a scan based on differnt priorities. Forms a Producer-Consumer relationship with the Visualizer. Once launced it will continue to produce network trees(see below) for the buffer until it recieves a pause/stop command, or completes a scan of the network.
2. Composite: Following the composite design pattern, this object represents the hierarchial Network tree, either in part in whole.
3. Threaded sub-Scanners: Each logial part of an AWS network has an assigned threaded scanner to it. Depeding on what is a priority to scan first the scanner will launcha number of threads for each part wich in turn will launch its on sub scan threads based on perfromance requirments. Each threaded SubScanner will scann 100 of its children,place in buffer and then contineu to scan another 100.
4. Smart buffer: The link between the Visualizer and the Scanner as wel las the "brain" of the system. The individual scanner threads wil ladd to the buffer

as soon as they have finished constructinga tree then go back to scanning. The trees arrive disjoint and posisbly overlap. The smart buffer constructs the entire network tree from the disjoint trees it recieves accurately. It stores the moat upto date construct for the Visualizer.

5. REST API and Server: The REST server joins the backend to the front end. See architecture below.

The client layer consists of a Single page Application and the Visualizer. The SPA is tee user interface with all the scanning options and controll. and the visualized graph and scanned information. The Visualizer polls on the REST server requesting latest tree from the smart buffer, rendering it and presenting it to the user.

RESTFUL Design:

User interaction is mapped from the user interface onto diffennt API calls on the server. Each call launches a backend method to fullfill the request. The API calls are:

1. POST, ConnectToAccount: Takes the users acces and private key and creates a conenction to AWS for the remainder of the session .
2. GET, ScanNetwork: Launches a scan of the entire network, Visualizer polls on results as they stream in.
3. GET, StopScan: Cancels current scan.
4. GET, PauseScan: Pauses current scan.
5. GET, ResumeScan:Resumes current scan Visualizere polls on buffer.
6. GET, ScanFrom: Performs a subscan starting from the given point and scanning around its general vicinity.
7. GET, ScanUP: Scans the next logical area above what the ScanFrom scanned.
8. GET, ScanDown: Scans the next logical area below what the ScanFrom scanned.

Threading Tactic: Each network component Region,VPC,Subnetwork,Instance has a threaded scanner associated to it. it si up to the Scanner depending on what the user is searching for to construct the scan from this subscanners. By defualt The scanner will thread a RegionScanner for each region and each one will have one scanner for the other parts. However if the scan is narrowed then scanner will thread on the most cost and performance effective way to compelte the scan . For example if the scan was to only scan a single VPC then other Regions would not need to be scanned and thus only one RegionalScanner will be launced, one VPCScanner will be launced and it with thread on collections of its subnetworks. We trying to keep number of threads needed to a mninimum, but they are invaluabel to speeding up the scanning.

5 Functional Requirements

This section specifies the functional specifications for the AWS Network Visualizer system. It defines the user-system interaction and relationship between users and the product. It will provide the expected functionality for all user cases as well as the activity processes for the system.

5.1 Description of AWS

Amazon web services provide a cloud based service for hosting a clients network. There is a lack of information regarding ones network, specifically the logical representation on the system for the client to make sense of their network. The network visualizer aims to improve clients understanding of their own network, how AWS works and possible insights in the managing of their network.

5.2 Required Functionality

The system must be able to:

1. Be accessible to registered, valid AWS customers.
2. Scan the networks located in different regions.
3. Provide an interactive hierarchical and visual representation of the networks.
4. Give additional information on the network statistics, construction, etc.
5. Provide a clear image of the clients' virtual networks.
6. Improve the AWS clients' experience.

5.3 Use case Prioritization

5.4 Use case/Service contracts

Use Case	Pre Condition	Post Condition	Description
Viewing the Hierarchy	The application must be connected to the server and able to read in network data. The server must be active and have acess to AWS.	The web page continuosly updates with the new data being loaded.	This use case forms the core of the project, as the primary purpose of the application is to visualise the structure of the virtual network.
Select a region	The application must be connected to the server and able to read in network data. The server must be active and have acess to AWS.	The previous region's visualisation is replaced by the selected region.	There are a number of AWS regions. For simplicity's sake, the visualiser only visualises one at a time, but allows for switching betweeen them.
Zooming in/out. Moving the hierarchy	The application must be connected to the server and able to read in network data. The server must be active and have acess to AWS.	The hierarchy's position or level of zoom is altered.	Since the visualised network may be large, the user can move it about and zoom in, for a better view.
Clicking a node.	The application must be connected to the server and able to read in network data. The server must be active and have acess to AWS.	The selected node is highlighted. Node related information is displayed below the hierarchy.	Each node in the network has a number of attributes that can provide valuable information. This way, the information can be displayed in a neat manner.
Hovering over a node.	The application must be connected to the server and able to read in network data. The server must be active and have acess to AWS.	Edges connecting nodes of different levels are rendered invisible. Edges connecting nodes on the same level are made vidible.	It is possible for nodes on the same level of the network to have relationships. They are normally hidden, in order to present a cleaner hierarchy.

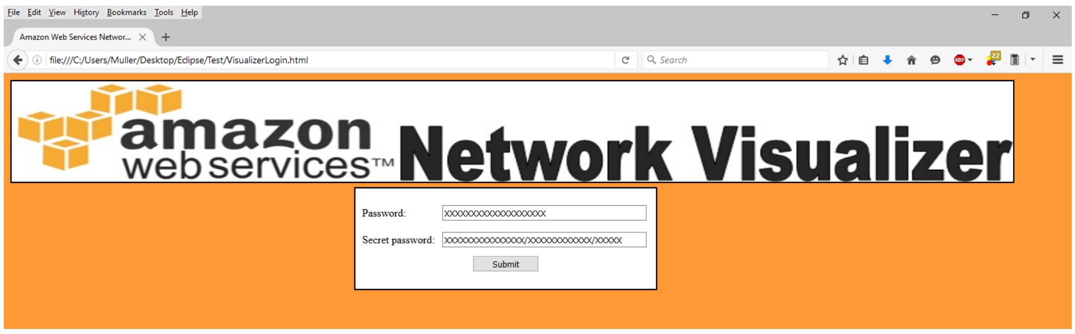
6 Specification Update

7 Using the system

The functionality of the web page will be spread between the following use cases:

7.1 Login

This is the first web page of the visualizer. The user is presented with two fields: One for the user's normal AWS passoword and another for their secret AWS password. The user then enters the requested details and the server is queriedm to verify the user's identity. Only when the correct passords are supplied, will the user be able to continue to the visualizer.

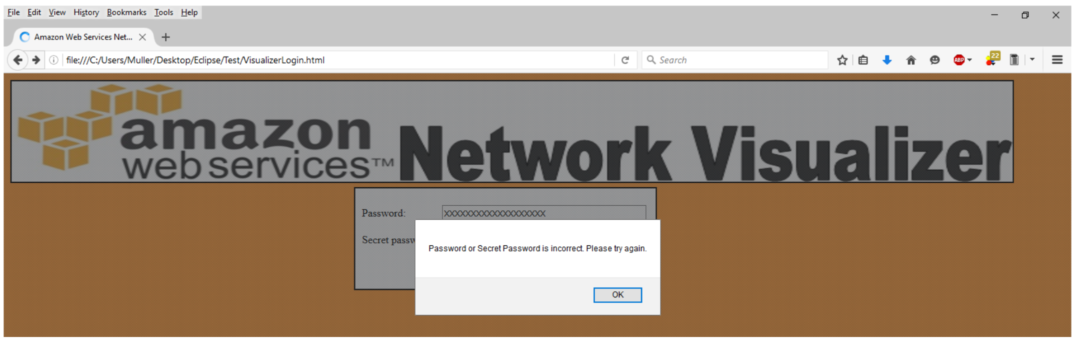

file:///C:/Users/Muller/Desktop/Eclipse/Test/VisualizerLogin.html

Password: xxxxxxxxxxxxxxxxxxx

Secret password: xxxxxxxxxxxxxxx/xxxxxxxxxxxxx/xxxxx

File Edit View History Bookmarks Tools Help
Amazon Web Services Networ...
Search
amazon web services™ Network Visualizer
Submit

If the user provides incorrect information, they presented with a message indorming them of this ocurrence.


file:///C:/Users/Muller/Desktop/Eclipse/Test/VisualizerLogin.html

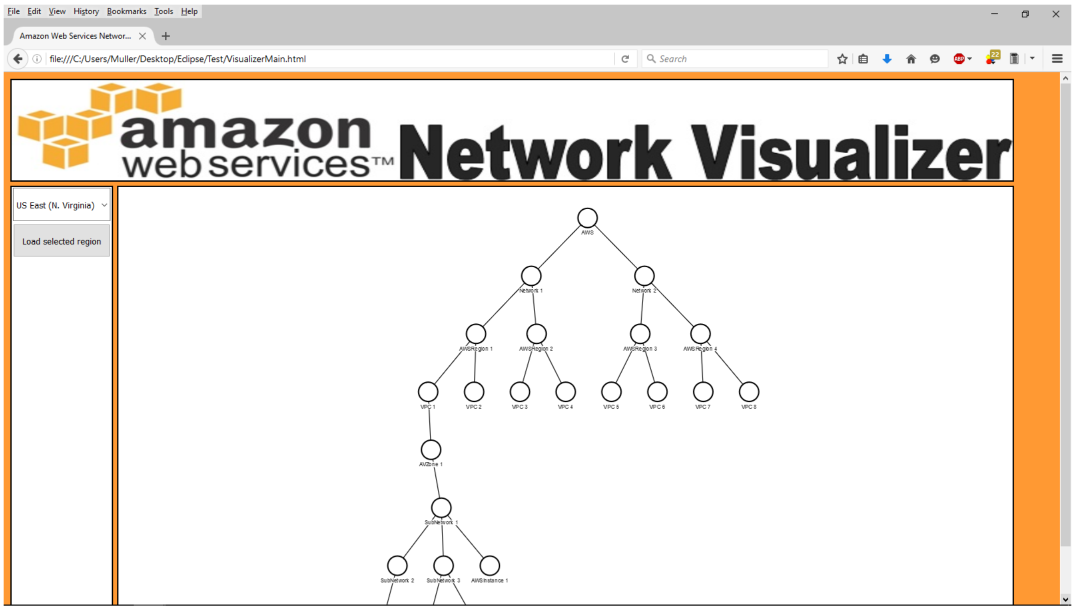
Password: xxxxxxxxxxxxxxxxxxx

Password or Secret Password is incorrect. Please try again.

File Edit View History Bookmarks Tools Help
Amazon Web Services Net...
Search
amazon web services™ Network Visualizer
Secret passw
OK

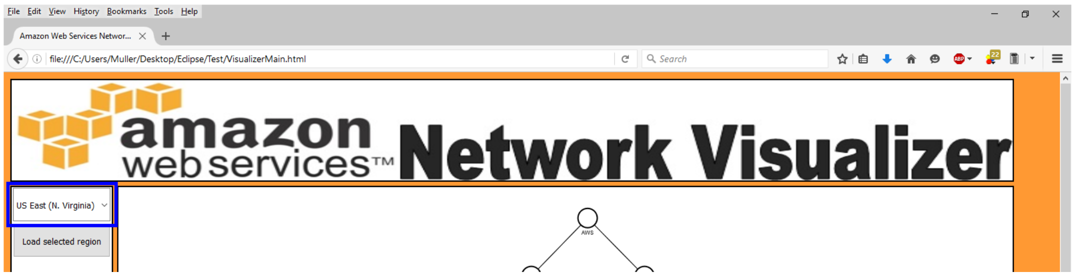
7.2 Visualizer

This use case pertains to the network visualization. The network is presented in a hierarchial manner, to make traversing the tree easier.

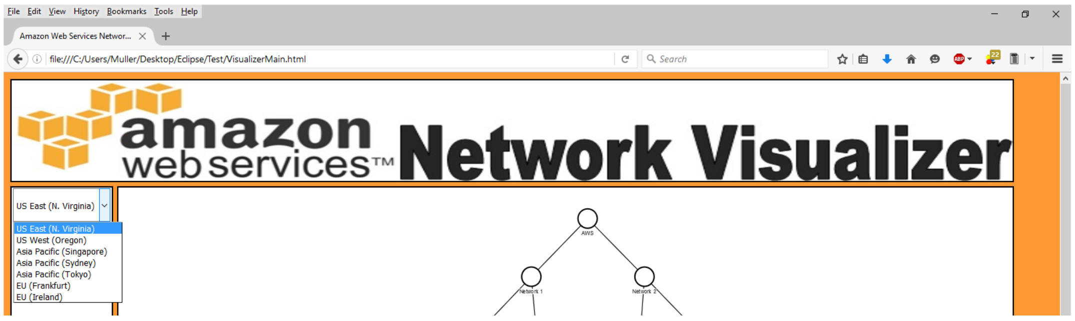


7.3 Regions

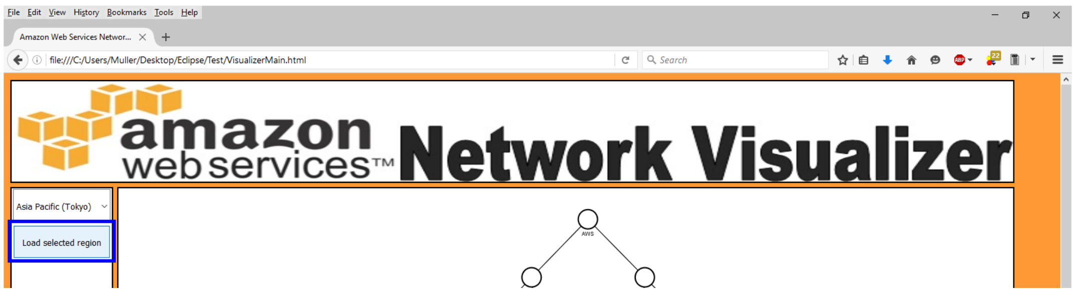
The dropdown menu in the left column displays the active region's name. The active region is the one that is shown in the visualizer.


File Edit View History Bookmarks Tools Help
Amazon Web Services Networ...
file:///C:/Users/Muller/Desktop/Eclipse/Test/VisualizerMain.html
Search
amazon web services™ Network Visualizer
US East (N. Virginia)
Load selected region
AWS

Clicking the menu will display a list of the official AWS regions. Clicking one of the options will set the current region.

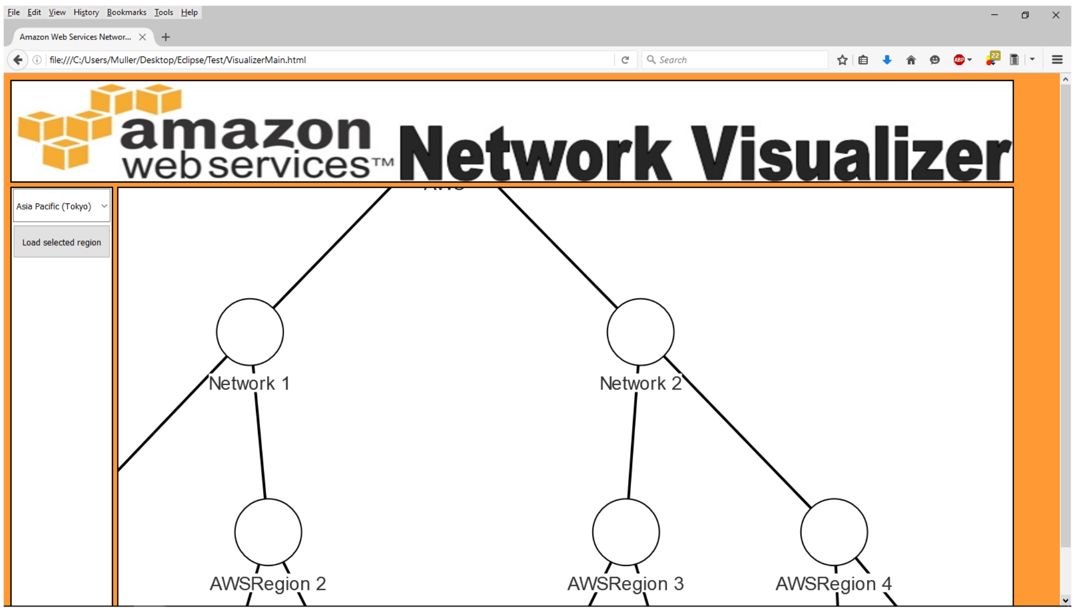


Clicking the "Load selected region" button sends a request to the server, which will then load the selected region and return a web page with the selected network representation


File Edit View History Bookmarks Tools Help
Amazon Web Services Networ...
file:///C:/Users/Muller/Desktop/Eclipse/Test/VisualizerMain.html
Search
amazon web services™
Network Visualizer
Asia Pacific (Tokyo)
Load selected region
AWS

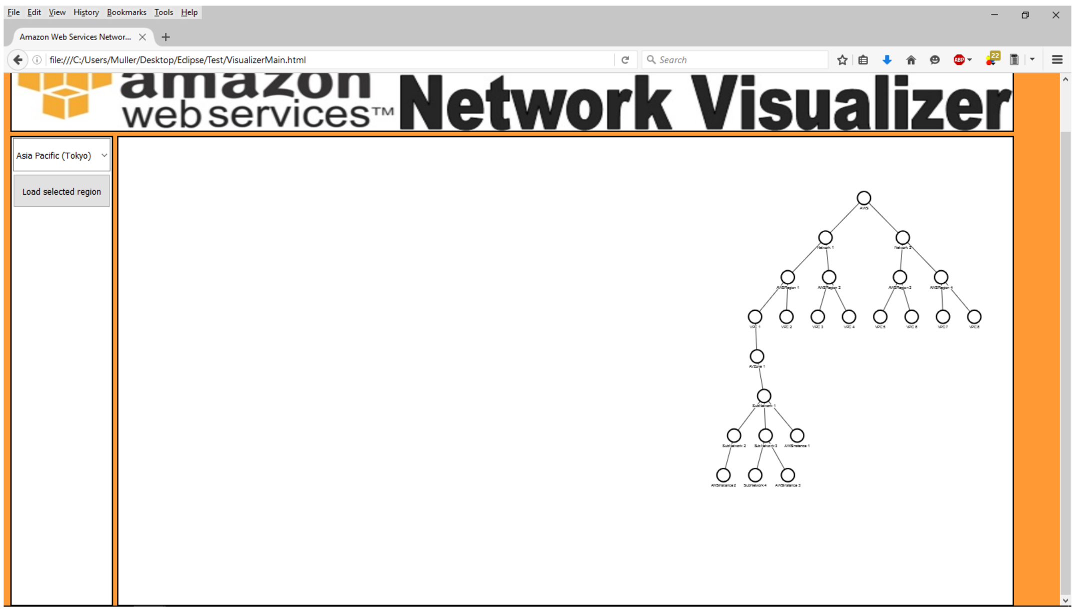
7.4 Zooming

Hovering the mouse over the visualizer's window and rolling the mouse wheel will change the zoom of the visualizer.


File Edit View History Bookmarks Tools Help
Amazon Web Services Networ...
file:///C:/Users/Muller/Desktop/Eclipse/Test/VisualizerMain.html
Search
amazon web services™
Network Visualizer
Asia Pacific (Tokyo)
Load selected region
Network 1
Network 2
AWSRegion 2
AWSRegion 3
AWSRegion 4

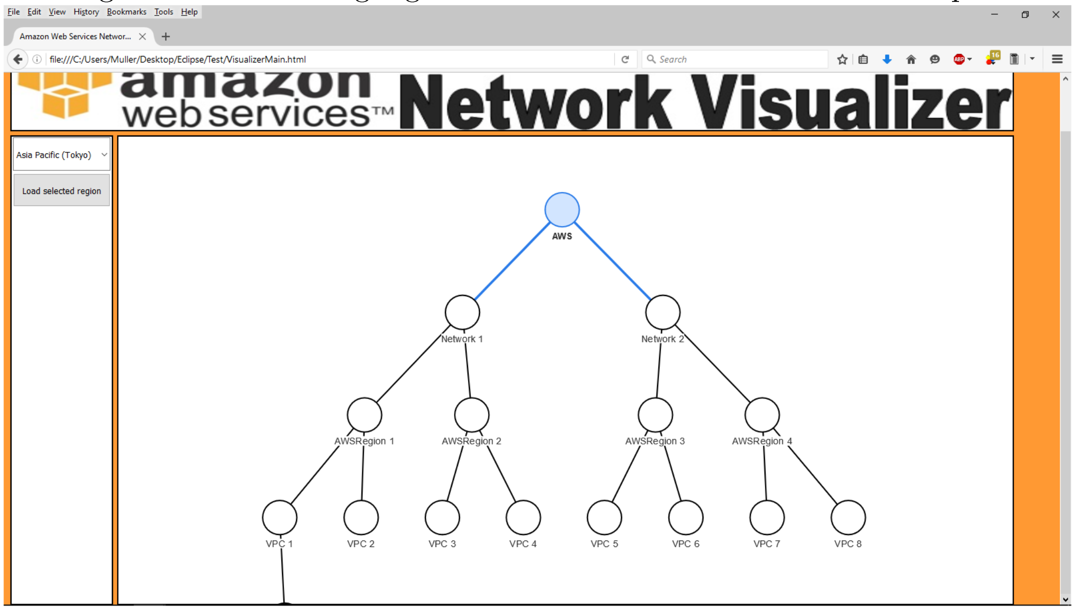
7.5 Moving

Clicking and dragging inside the visualizer's window will move the image.


File Edit View History Bookmarks Tools Help
Amazon Web Services Networ...
file:///C:/Users/Muller/Desktop/Eclipse/Test/VisualizerMain.html
Search
amazon web services™ Network Visualizer
Asia Pacific (Tokyo)
Load selected region

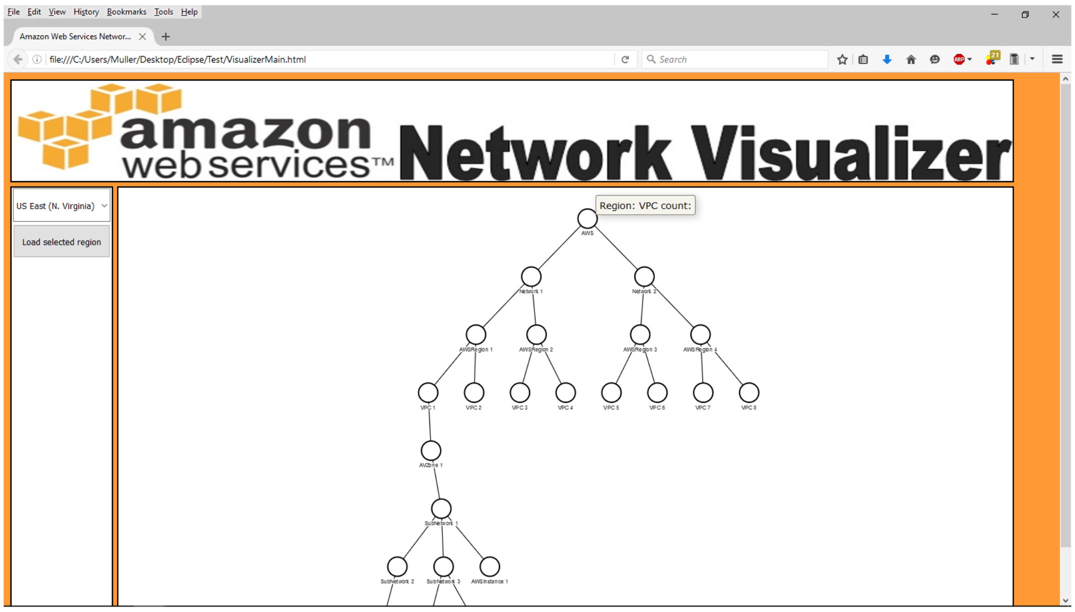
7.6 Selection

Clicking on a node will highlight the selected node and its relationships.


File Edit View History Bookmarks Tools Help
Amazon Web Services Networ...
file:///C:/Users/Muller/Desktop/Eclipse/Test/VisualizerMain.html
Search
amazon web services™ Network Visualizer
Asia Pacific (Tokyo)
Load selected region
AWS
Network 1
Network 2
AWSRegion 1
AWSRegion 2
AWSRegion 3
AWSRegion 4
VPC 1
VPC 2
VPC 3
VPC 4
VPC 5
VPC 6
VPC 7
VPC 8

7.7 Information

Hovering the mouse over a node will display a small box, specifying its name and the number of VPC's it has.


File Edit View History Bookmarks Tools Help
Amazon Web Services Networ...
file:///C:/Users/Muller/Desktop/Eclipse/Test/VisualizerMain.html
Search
amazon web services™ Network Visualizer
US East (N. Virginia)
Load selected region
Region: VPC count:
AWS
Network 1
Network 2
AWSRegion 1
AWSRegion 2
AWSRegion 3
AWSRegion 4
VPC 1
VPC 2
VPC 3
VPC 4
VPC 5
VPC 6
VPC 7
VPC 8
AWZone 1
SubNetwork 1
SubNetwork 2
SubNetwork 3
AWSInstance 1