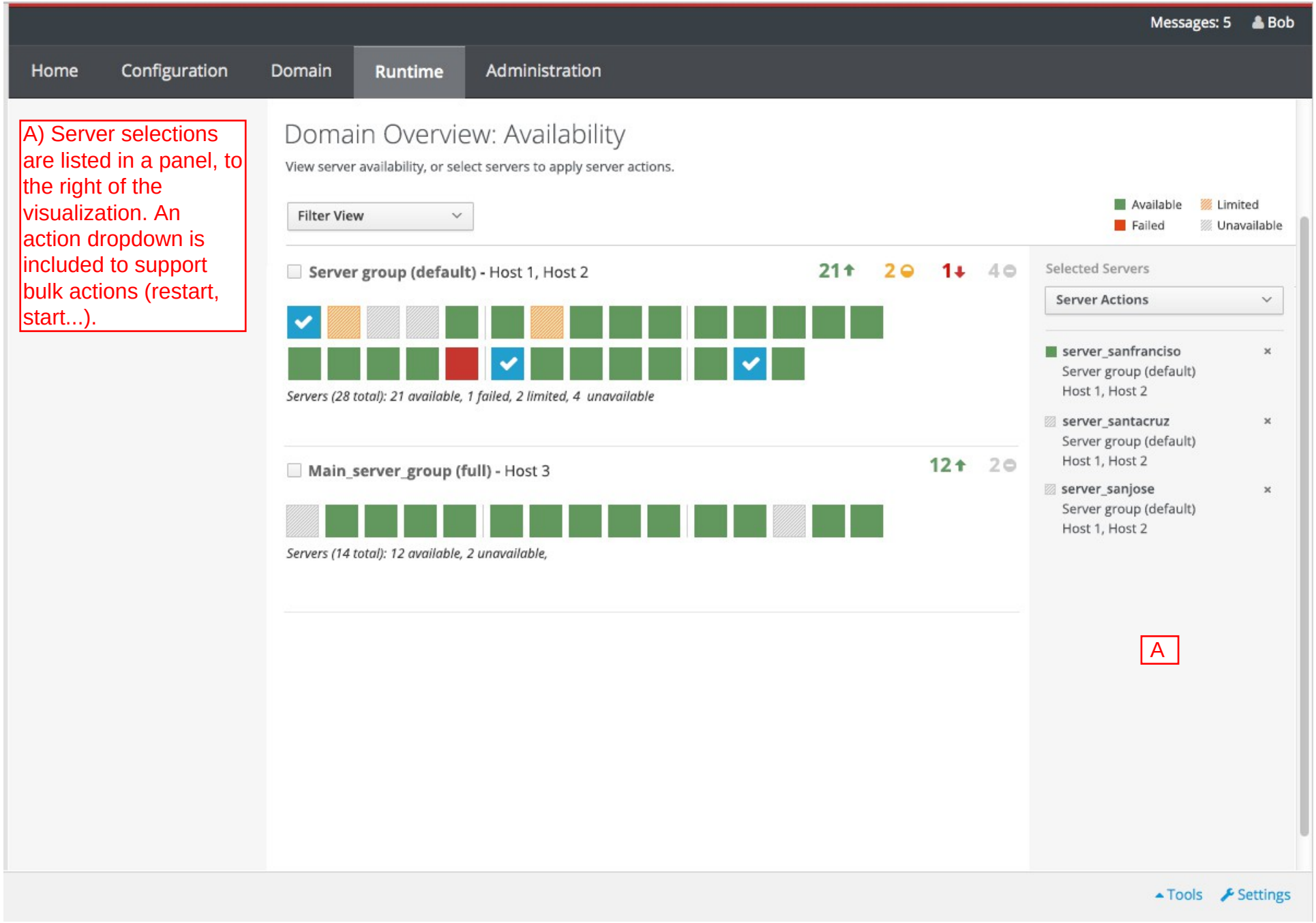
Domain Overview

Redesign proposal 02 - DRAFT
“Selection & bulk actions” version
January 05, 2015

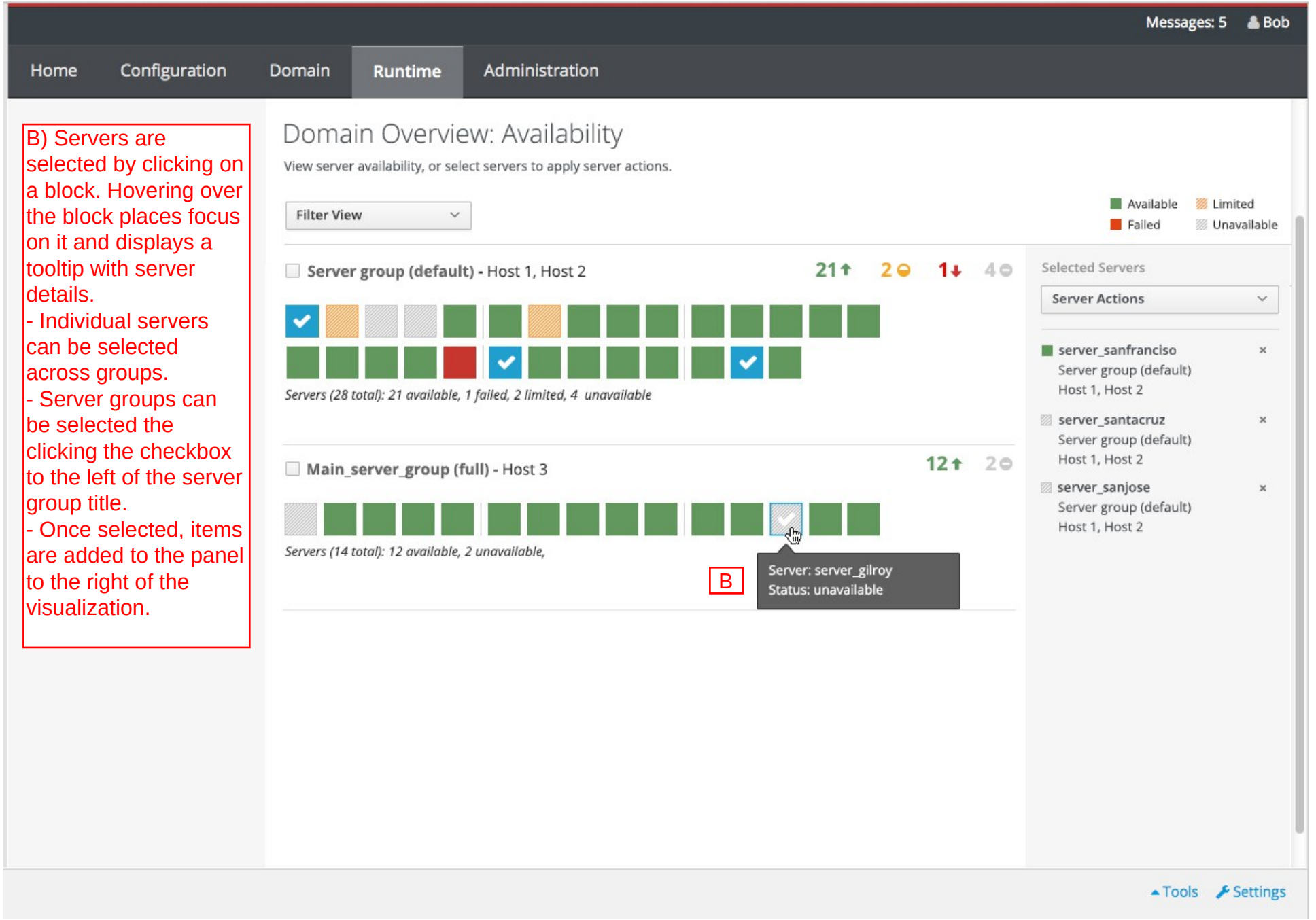
Changes include:

- For the moment, removed the alerts tab to focus on availability status.
- Added a status rollup for each server group.
- Added checkbox selection for server groups.
- Added inline selection for servers, across groups.
- Added a right panel to provide a selection summary and a bulk-actions dropdown.
- Simplified the info tooltip (on hover).

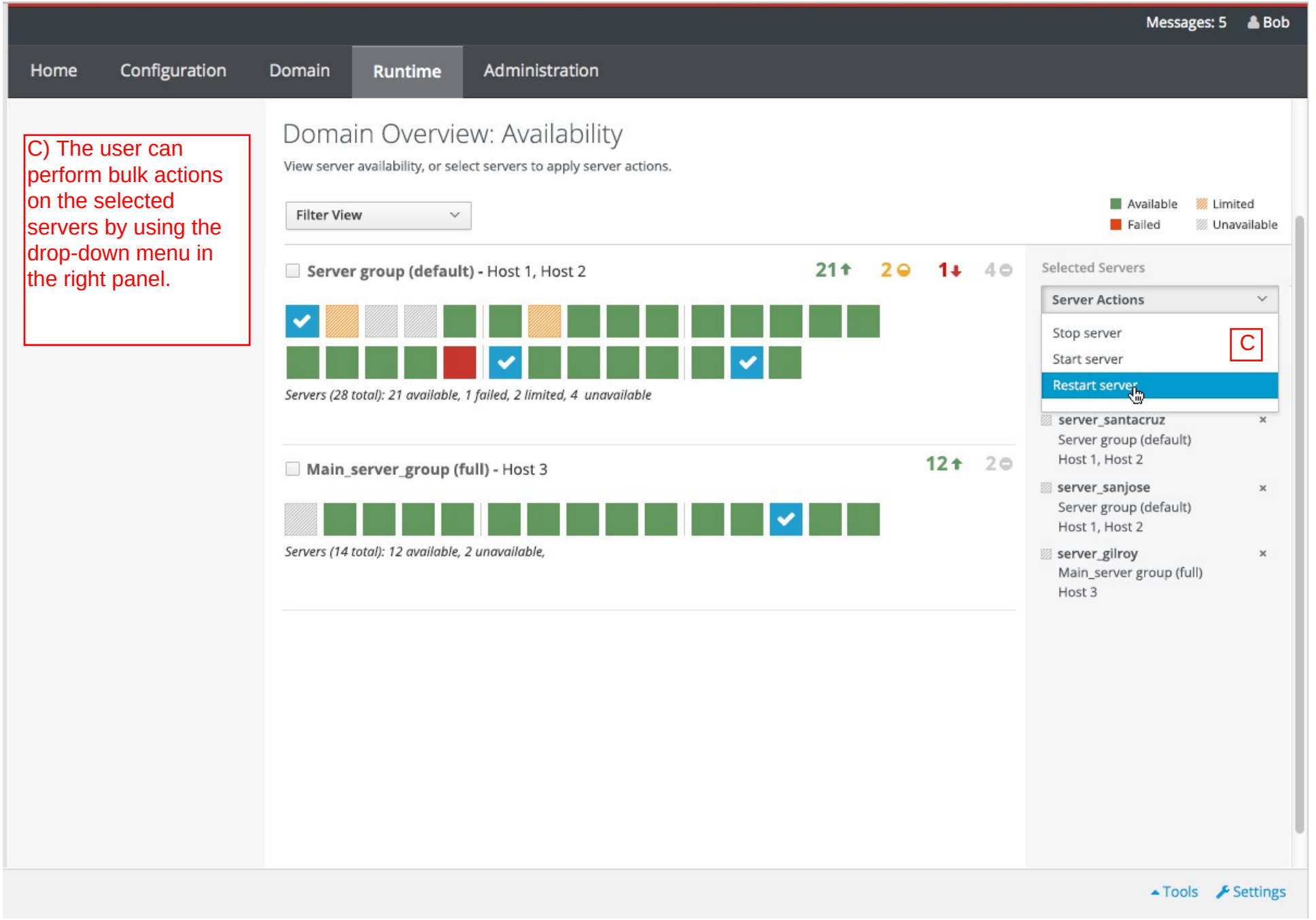
Availability view, with selection and bulk actions



Hover info-tip



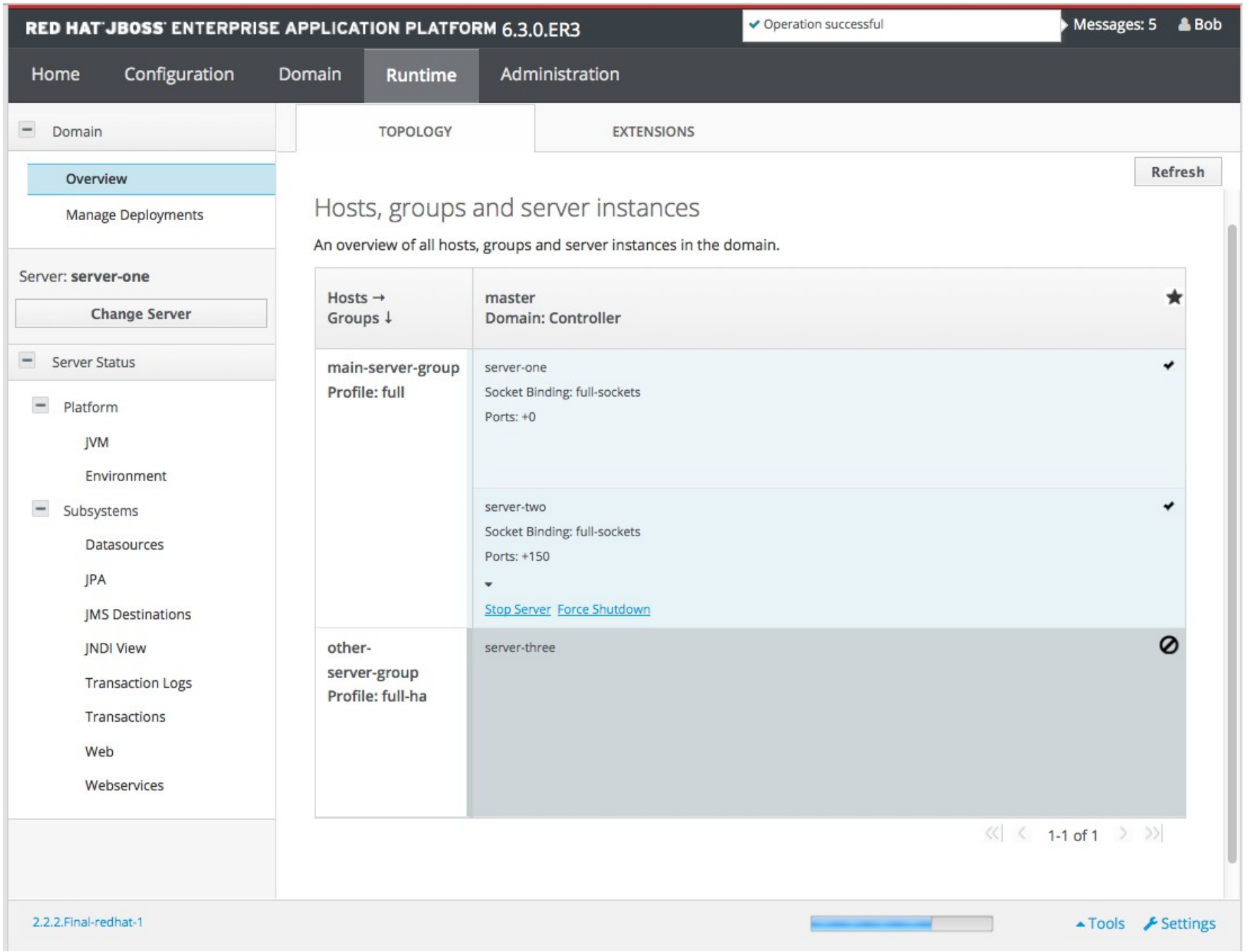
Bulk actions dropdown



Domain Overview

Redesign proposal 01 – DRAFT
Heatmap visualization
July 01, 2014

Current (EAP) Domain Overview



Stated Goals

1) Primary

- Provide a (visual) **overview of a domain**, *which could possibly include over 100 nodes running 10s of server groups.*
- The goal of the visualization is to represent data relevant to **monitoring** system health.

2) Secondary

- Provide a means to **drill-down** through the domain overview, to get more granular info.
- Allowing the user to **take action where possible** (provide links to start "tasks," similar to home pg model).

Sub-task requirements (info contributed to-date):

Information to represent:

Monitoring:
- Server **Availability** (started/stopped).

- [Representation of relationship] **server group- host**

- **Alerts** (red, green, yellow.)
Particularly "runtime" and configuration changes.

- Additional suggestions:
  - Application deployment info (which apps deployed and where)
  - Database connectivity (connected/ not connected).

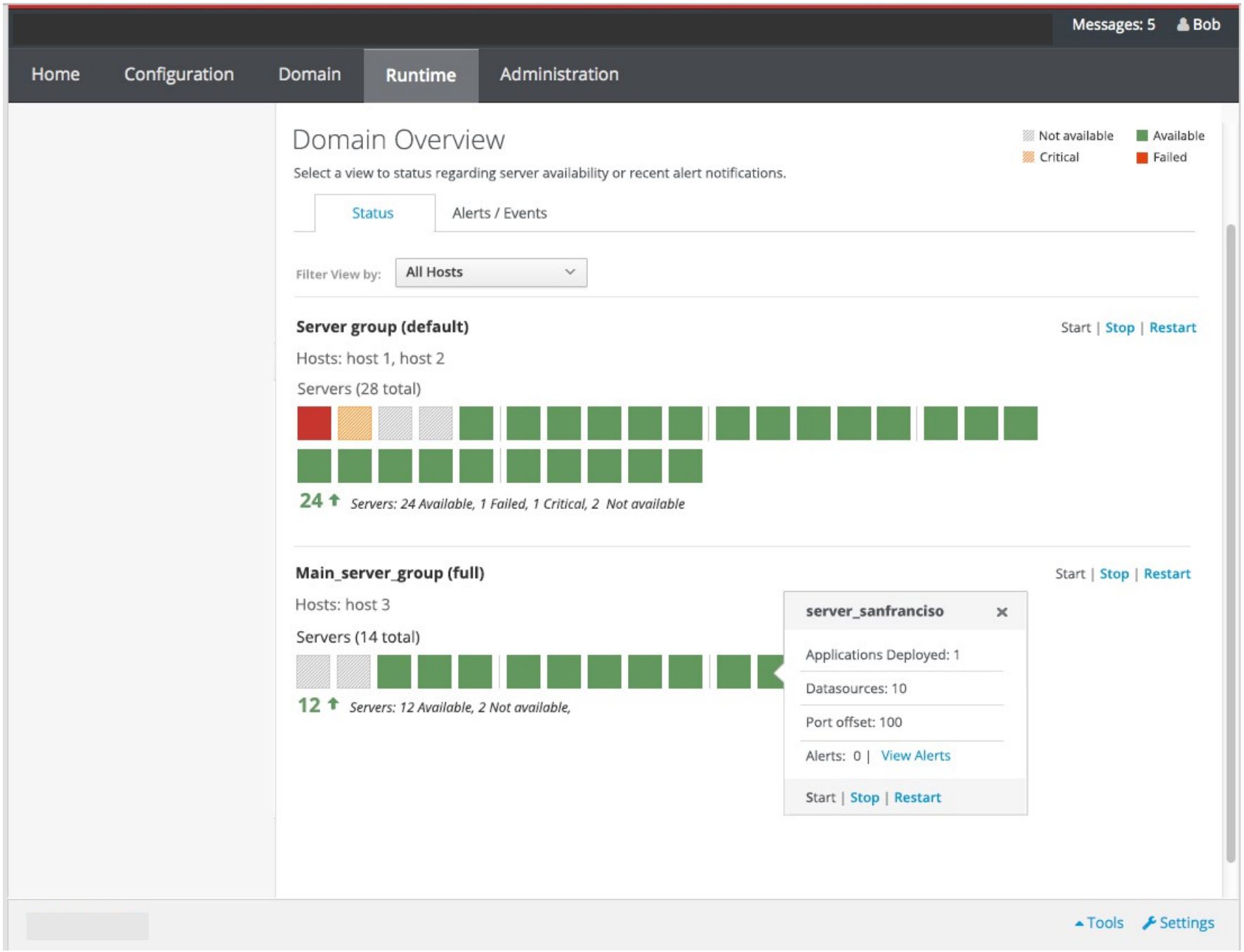
How to represent, considerations:

a) Provide drill down to details
"...provides quick access to specific information and reduces cognitive and technical load." "...Further information after drill down, such as: JVM status (memory, etc), ports, deployments and etc."

b) Support data comparison
"...large amounts of data w/o means to compare specific attributes across hosts, groups or servers doesn't make much sense."

c) Flag alerts
"Considering a) and b) … a mechanism that says "this entity requires attention" with the ability to get to specific information on a second level probably works best."

Current Proposal



Current Proposal

1. Status Tab:
View overall availability (health) of servers.

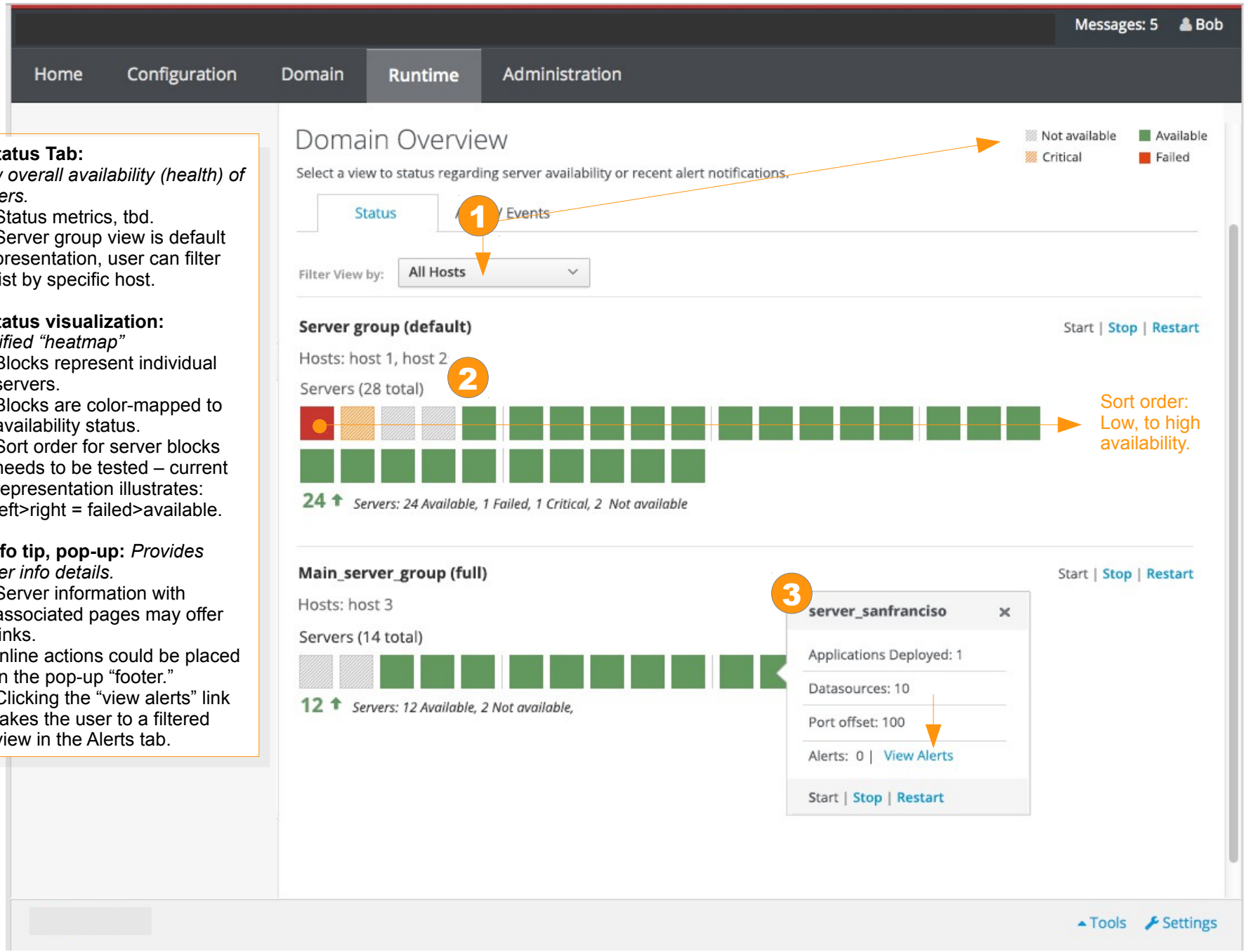
- Status metrics, tbd.
- Server group view is default presentation, user can filter list by specific host.

2. Status visualization:
Modified "heatmap"

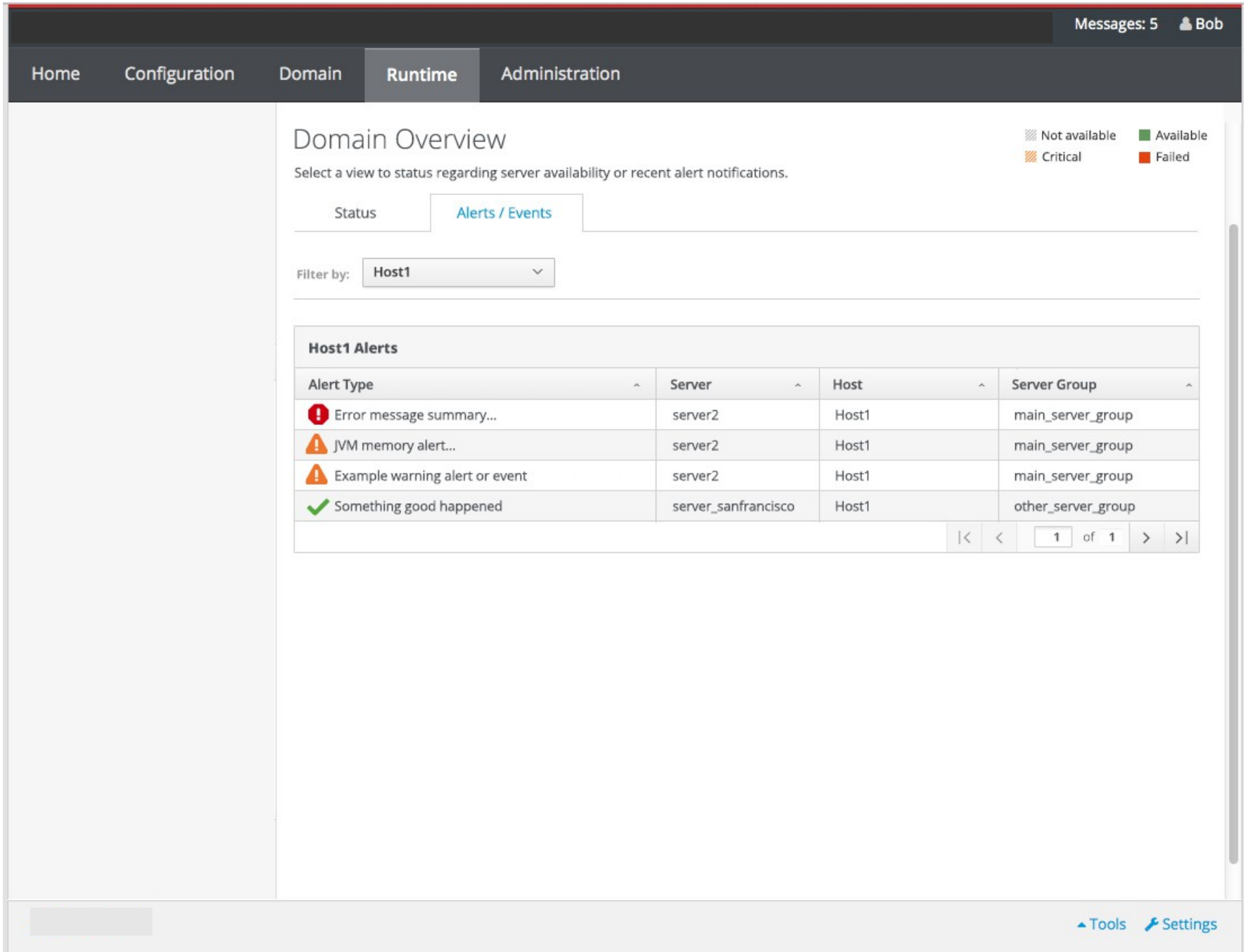
- Blocks represent individual servers.
- Blocks are color-mapped to availability status.
- Sort order for server blocks needs to be tested – current representation illustrates: left>right = failed>available.

3. Info tip, pop-up: *Provides server info details.*

- Server information with associated pages may offer links.
- Inline actions could be placed in the pop-up "footer."
- Clicking the "view alerts" link takes the user to a filtered view in the Alerts tab.



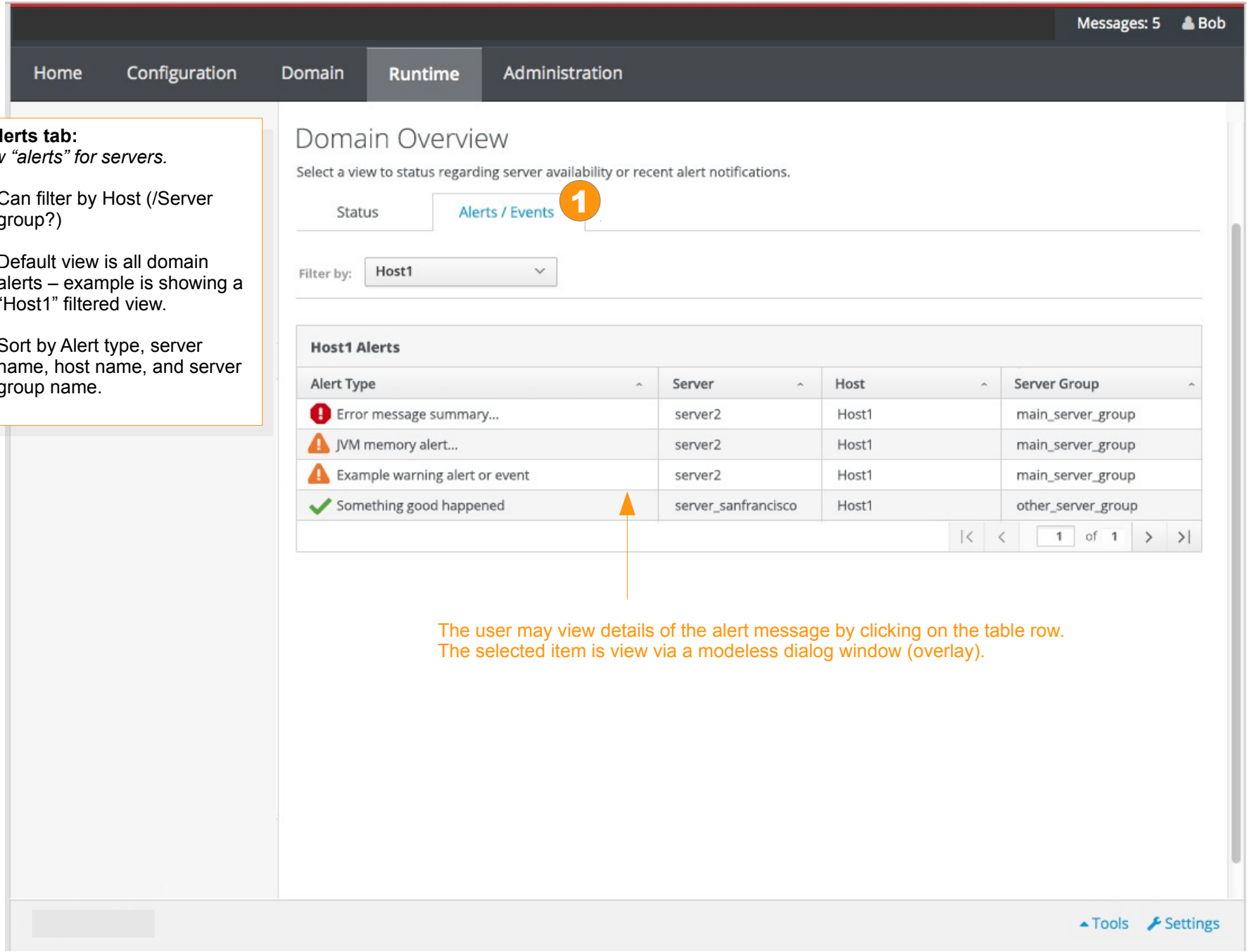
Possible – Adding Alert/Events Tab



Possible – Adding Alert/Events Tab (persistent tab tbd)

1. Alerts tab:
View "alerts" for servers.

- Can filter by Host (/Server group?)
- Default view is all domain alerts – example is showing a "Host1" filtered view.
- Sort by Alert type, server name, host name, and server group name.



The user may view details of the alert message by clicking on the table row.
The selected item is view via a modeless dialog window (overlay).

Questions

What are the states for server availability?

- Not available (taken down, not tied to issues).
- Available (running as expected)
- Failed (stopped due to errors or failures)
- Critical (Question: is there a state between available and failed, and if so how is that calculated?)

What is the relationship between Status and Alerts?

- Do “x” alerts equate to a change in availability status, or can they function independently. Can you have a error on a server and it still is “available?”

Status visualization: Order of servers (blocks) within the server groups “heatmap.”

- Should the servers be laid out in the visualization by level of availability/status (as illustrated), or by some other ordering (A-Z, Z-A...)?
- If by server name or etc., will the user be able to memorize location? Plan to follow-up by testing.

How does the Alerts tab fit in with the current Notification message queue?

Design Research

Initial explorations (known/standard visualizations):

Heatmaps

A heat map is a graphical representation of data where the individual values contained in a matrix are represented as colors.

Pros:

- Can provide visualizations for very large data sets.
- Can be used to show high-level monitoring metrics (availability and etc.)
- There are Open Source solutions/packages available.

Concerns:

- There's a learning curve to reading the display, if unfamiliar.

Other possibilities (TBD):

Dashboard

Single page, real-time user interface, showing a graphical presentation of various high-level metrics to enable instantaneous and informed decisions at a glance.

Concerns:

- Don't have enough aggregated data to display.
- Would need a fair amount of dashboard widgets (various graphs and etc) to make the display useful.

Tiles (image tiles, playing cards)

Grid display of high-level data, typically modular and more visual in nature.

Concerns:

- If used for text data, it's difficult to scan/parse.
- Difficult to represent relationships/comparisons – it's basically a flat list.

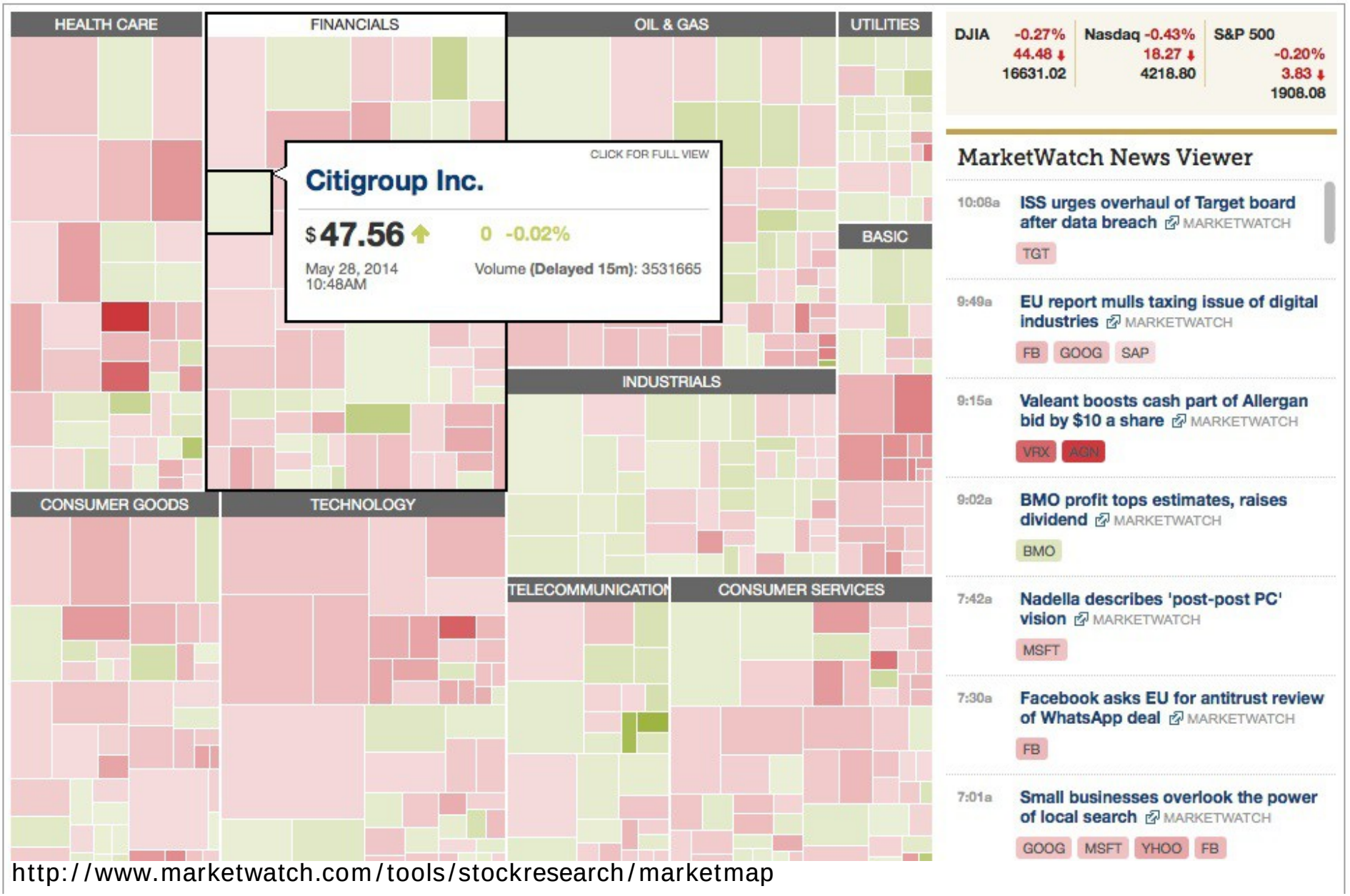
Topology Maps

Visual representation of a network, using representations for nodes and connections.

Concerns:

- The layout algorithm is too complex
- It might not scale well.

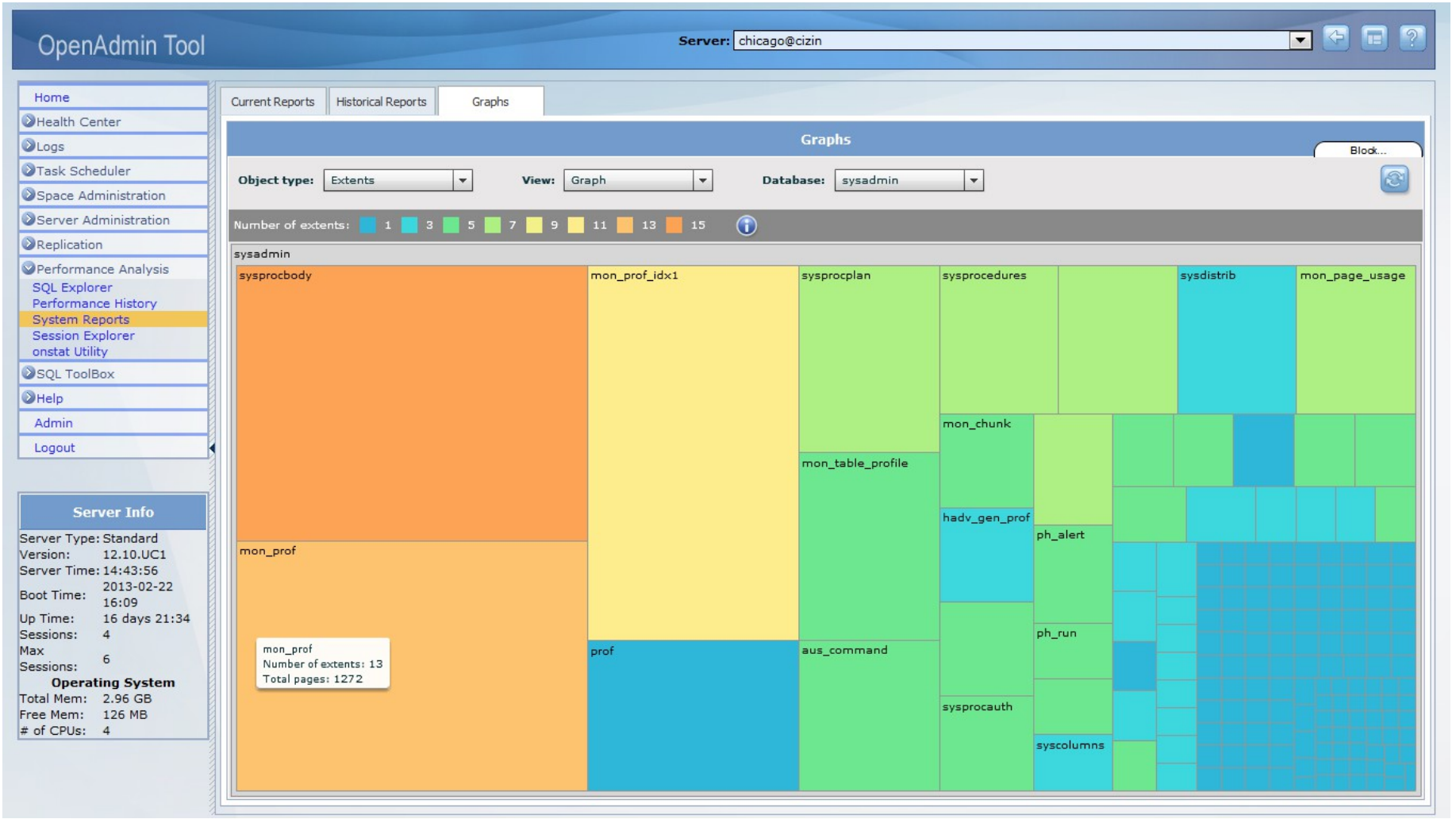
Heat Maps are well suited for scale and visualizing values for large data sets



http://www.marketwatch.com/tools/stockresearch/marketmap

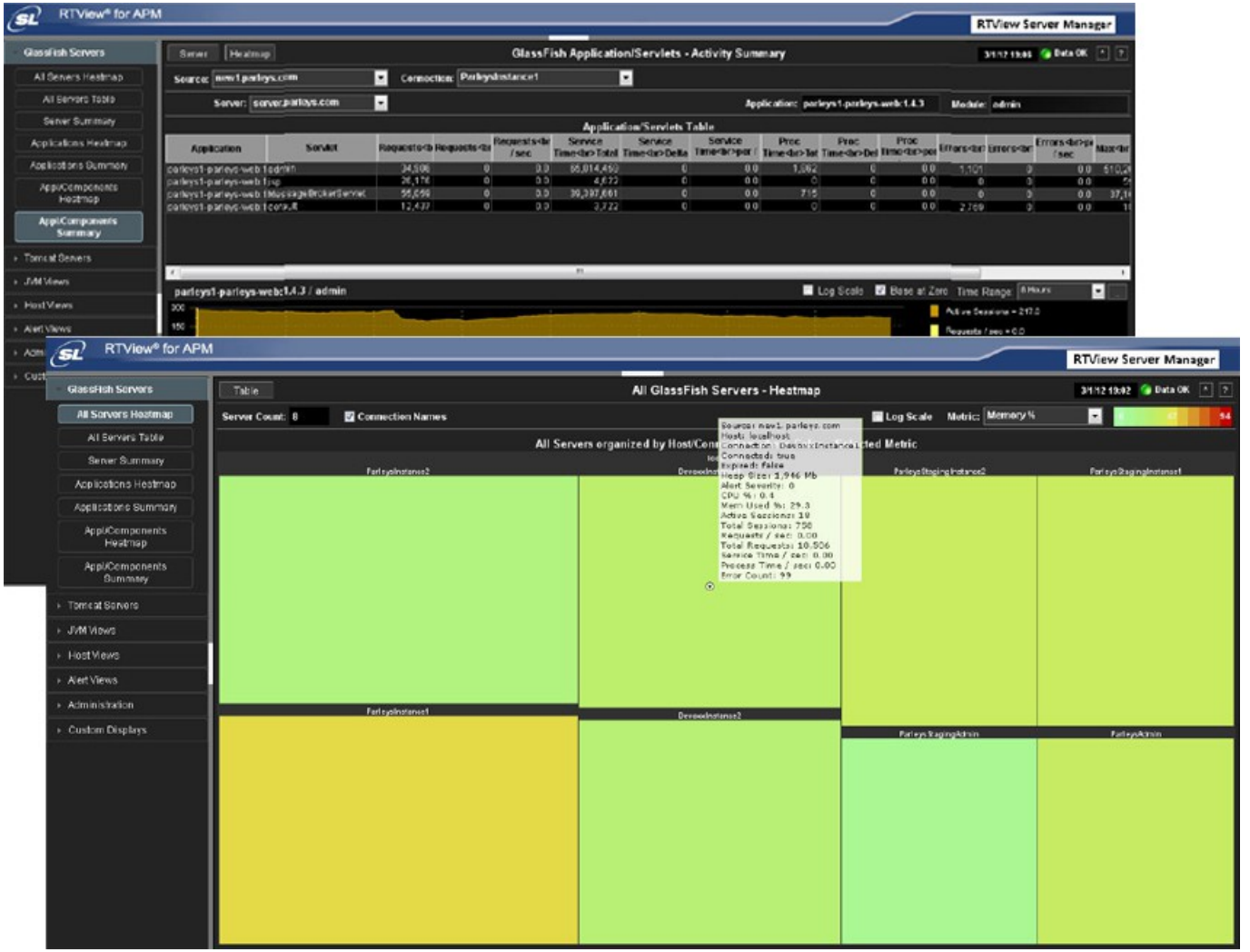
A heat map is a graphical representation of data where the individual values contained in a matrix are represented as colors. A Heat Map is a type of treemap that is well suited for visualizing large flat data volumes. They both use a similar system of color-coding.

Heat Maps applied to system applications



OpenAdmin Tool: Uses Heat Map Graphs to monitor Extent and Buffer Pool Usage

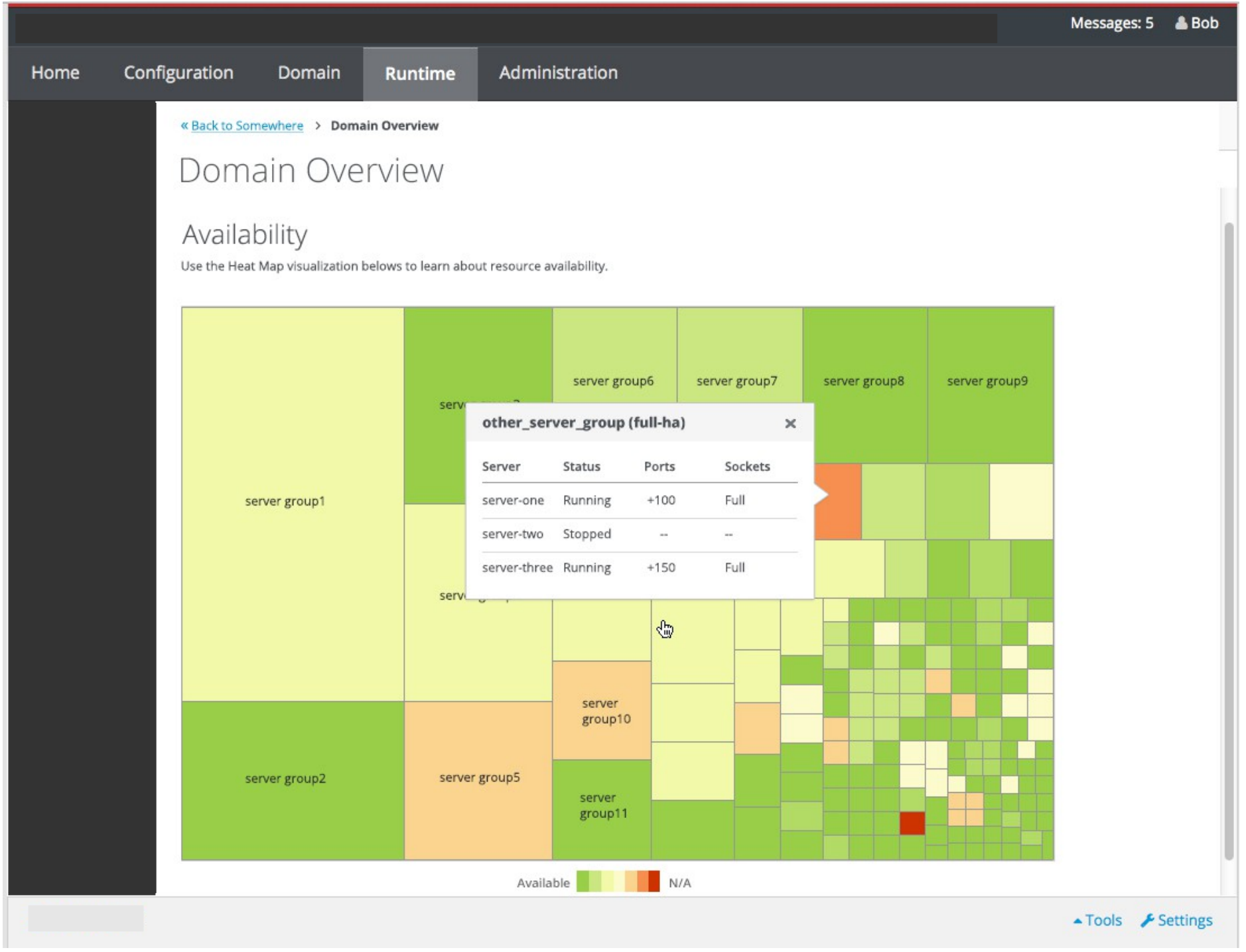
Heat Maps applied to system applications



RTView EM Solution Packages: “The GlassFish Solution Package provides a high level Application Activity heatmap for a high level, real-time health summary of your entire GlassFish environment.”
http://www.sl.com/products/rtviewem_glassfishservers.shtml

Initial design iterations (heatmap only)

First draft heatmap – Server Group view for “availability”



Second Iteration: Stacked heatmap