ScaleArc for SQL Server 3.8.4

Administrator's Guide

Trademarks

© ScalArc, Inc. 2016

© ScaleArc 2016

ALL RIGHTS RESERVED.

THIS DOCUMENT IS PROPRIETARY TO SCALARC, INC., AND MAY NOT BE REVISED, MODIFIED OR OTHERWISE ALTERED WITHOUT SCALEARC'S AUTHORIZATION. ALL REPRODUCTIONS OF THIS DOCUMENT MUST NOT REMOVE OR REDUCE THE VISIBILITY OF ANY PROPRIETARY NOTICES.

ScaleArc®, ScaleArc iDB®, and ScalArc® are registered trademarks of ScalArc, Inc and/or its affiliates in the U.S, India and certain other countries. Other names and marks used herein are trademarks or service marks of their respective owners.

Disclaimer

The information disclosed in this document, including all designs and related materials, is the valuable property of ScalArc, Inc. ScaleArc reserves all copyrights and other proprietary rights to this document, including all design, manufacturing, reproduction, use, and sales rights thereto, except to the extent said rights have been expressly granted to others. The product(s) discussed in this document are licensed in accordance with the terms of ScaleArc's End User Software License Agreement. The information in this document is subject to change at any time, without notice.

Customer Support

Contact ScaleArc sales at sales@scalearc.com or call +1-408-837-2250. You can contact technical support at support@scalearc.com or call +1-855-800-SCAL(7225).

Contents

Chapter 1 : ScaleArc Overview 8
1.1 What is ScaleArc? 8
Chapter 2: Getting Started with ScaleArc 13
2.1 Upgrading from Previous Versions of ScaleArc 13
2.2 Installing ScaleArc 13
Chapter 3: Configuring Network Settings 14
3.1 Network Setting 14
3.2 Clearing events 16
Chapter 4: ScaleArc License 17
Chapter 5: ScaleArc Configuration 21
5.1 Network Settings 21
5.2 System Settings 24
5.2.1 System Status 24
5.2.2 System Config 25
5.2.2.1 System Config 25
5.2.2.2 Administrative User 25
5.2.2.3 Configuration 25
5.2.2.3 Date and Timezone 26
5.2.2.4 Security Settings 26
5.2.3 Services 27
5.2.4 Windows AD Setup 27
5.3 Alert Settings 28
5.4 High Availability Settings 29
5.5 License Settings 31
5.6 User Management 32

5.7 API 33
5.8 CPU Distribution 34
Chapter 6: Clusters 36
6.1 What is a Cluster? 36
6.2 Creating Clusters 36
6.3 Managing Clusters 41
6.3.1 ScaleArc Cluster Info 41
6.3.2 Status 41
6.3.3 Settings 53
6.3.3.1 Users & DBs 53
6.3.3.2 Transparent Cache 54
6.3.3.3 Security 59
6.3.3.4 Log Settings 60
6.3.4 Servers 62
Chapter 7: Live Monitor 66
7.1 System Stats 66
7.2 Clusterwise Stats 67
7.3 Logs 68
7.4 Historical Stats 69
7.5 Diagnostics 70
Chapter 8: Analytics 73
8.1 The Landing Page 73
8.2 Cluster Analysis 74
8.3 Four Quandrant Graph 75
8.4 Database Analysis 75
8.5 Comparative Analysis 76
8.6 SQL Structure 76
8.6.1 SQL Statement Analysis 76

8.7 Log Search 77
Appendix A: ScaleArc Deployment Checklist 78
Appendix B: ScaleArc Third-Party Software Dependencies 82
Appendix C: List of Ports on ScaleArc 88

Preface

About the Guide

Welcome to the ScaleArc for SQL Server 3.8.4 Administrator's Guide. This document describes how to configure, manage and deploy ScaleArc as a solution in your network.

Who Should Read this Guide

This guide is for system administrators and SQL administrators deploying ScaleArc on a virtual, physical machine or on the supported cloud platform. If you are an administrator that works with a managed services provider or in a cloud environment please refer to the ScaleArc Cloud Deployment Guide for the particular provider. For more information refer to the whitepapers located at **http://scalearc.com/resources/whitepapers**. Readers should be familiar with general TCP network fundamentals, your environment's SQL implementation and the applications that are supported by them.

Typographical Conventions

This guide uses different typefaces to indicate different kinds of information. The following table explains these typographical conventions.

Table 1. Typographical Conventions

	Meaning	Description
	Reference	This icon appears alongside a reference.
	Note	*This icon appears alongside a note.*
	Warning	This icon appears alongside a warning.
`Button`	User Interface	Look for or interact with the identified element.

References

Working with ScaleArc requires a fundamental working knowledge of SQL and TCP/IP fundamentals.

The following references are provided for readers who require more background knowledge before working with ScaleArc.

1. **SQL Standards**
2. **TCP/IP Fundamentals**
3. **MSSQL Standards**
4. **SQL Server Documentation**

Contacting ScaleArc

Contact ScaleArc sales at sales@scalearc.com or call +1-408-837-2250. You can contact technical support at support@scalearc.com or call +1-855-800-SCAL(7225).

Chapter 1 : ScaleArc Overview

1.1 What is ScaleArc?

ScaleArc is the leading provider of database load balancing software. The ScaleArc software inserts transparently between applications and databases, creating an agile data tier that provides continuous availability and increased performance for all apps. With ScaleArc, enterprises also gain instant database scalability and a new level of real-time visibility for their application environments, both on prem and in the cloud. ScaleArc supports MySQL, Microsoft SQL Server, and Oracle databases.

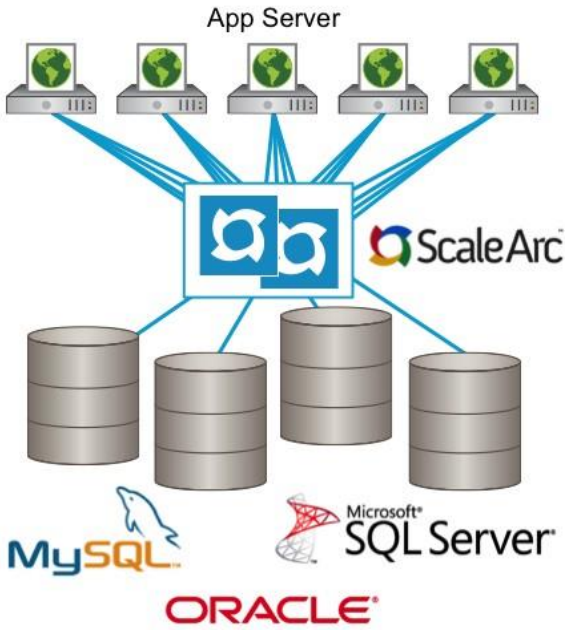


Fig 1.1

1.2 Benefits of ScaleArc

ScaleArc inserts transparently between applications and databases, requiring no modifications to the applications or to the databases to deploy.

Customers who deploy ScaleArc can take advantage of:

- Automated Failover/HA
- Zero Downtime Maintenance
- Database Scalability
- SQL Server Upgrades
- Improved App Availability and Performance
- Cloud Migration
- Real-time Operational Visibility

Automated Failover/HA

ScaleArc supports application - transparent automated database failover, shielding apps from downtime and preventing application errors. Operating at the SQL protocol layer, ScaleArc instantly and transparently routes database traffic around downed servers - even across multiple data centers for maximum availability - ensuring business continuity for SQL environments.

Zero Downtime Maintenance

ScaleArc delivers a seamless and highly reliable way to optimize traffic flows and reduce application error messages during planned downtime, such as:

- Patching database servers
- Applying software/security updates, including schema changes
- Performing storage maintenance
- Implementing master/slave failover
- Diagnosing potential performance issues

Database Scalability

ScaleArc enables database scale up instantly and database scale out transparently – with no changes to your app or database. The ScaleArc software drops in transparently between your apps and your database, routing database traffic and managing connections to the database with no app changes.

SQL Server Upgrades

Microsoft SQL Server 2012/2014 promise compelling new functionality, with AlwaysOn failover, synchronous replication technology, and the ability to read from secondary replicas. But the upgrade path to these new versions can be expensive and time consuming.

ScaleArc lets you start to benefit immediately from SQL Server 2012/2014's distributed capabilities – all without changing a single line of code in your critical business applications.

Improving App Availability and Performance

Due to its unique place in the network, ScaleArc has the ability to provide an unparalleled, in-depth view into an application's behavior with its database stack and provide you with the tools needed to take corrective measures to alleviate performance bottlenecks.

App performance bottlenecks, as related to databases, tend to fall into three main categories.

- Poor query performance
- Lack of concurrent capacity to handle high user loads
- Connection pooling / management issues

ScaleArc provides you with the analytics tools to pinpoint when each of the above problems is occurring, in real time. With ScaleArc's live monitoring, you can see how many database connections and queries your application is requesting and whether your database server/s are capable of handling the load.

Cloud Migration

Moving your workload from the premise to the cloud requires careful planning.

Without shared storage or the ability to migrate IP addresses between machines, traditional, on-premise high availability (HA) methodologies for SQL database failover don't work in the cloud. And – with less RAM, less CPU, and less storage available – the cloud can reduce performance for each database instance.

ScaleArc solves these problems, ensuring high SQL availability and performance through:

- Automatic failover in the cloud – delivered transparently across availability zones to instantly and automatically redirect traffic flows, ensuring business continuity and minimizing disruptions
- Seamless aggregation of capacity across multiple machines – balancing load across very large clusters to support significantly higher loads in the cloud and increase performance

Real-time Operational Visibility

ScaleArc collects extensive data regarding connections, user logins, SQL queries being executed, and much more - all of them in ScaleArc vs. the database and without sampling of data. ScaleArc's real-time analytics, centralized logging, and historical stats provide the tools you need to pinpoint issues and gain instant insights.

You can use these valuable analytics tools to:

- Track connection spikes that result in poor performance or outages
- Track a SQL session or SQL connection
- Pinpoint database performance issues, up to the minute

1.3 ScaleArc Features

The ScaleArc software has a variety of deployment options to match your database environment. ScaleArc is available as a software-only solution, as a virtual appliance for either a private cloud or a public cloud, or as a physical appliance.

Fig 1.2

Availability

- Automated Failover - With deep integration into SQL Server AlwaysOn technology, ScaleArc lets you harness the full capabilities of SQL Server 2012/2014 without any app changes. You gain automated failover across data centers, and by replacing the AlwaysOn Availability Group Listener as the primary destination of SQL connections, ScaleArc can:

  - Easily migrate load from one data center to another in the event of failure
  - Reduce the time for failover by automating many steps
  - Lower application errors that result during failover
  - Enable geo-aware load balancing for read traffic

- Surge Queue - Reduce application error messages by regulating load to prevent a database server from becoming overloaded or downtime due to crashes. ScaleArc's surge queue safely and effectively limits the load and holds load temporarily during a failover event with:

  - No change in query order
  - Transactional awareness
  - Two separate queues for reads and writes

- Read/Write Split - Boost performance for read-heavy applications and easily scale the database without fear of crashing with ScaleArc's read/write split capability. Without any application modification, ScaleArc automatically identifies which queries are reads and which are writes and transparently routes reads to the secondary and writes to the primary in a manner that is:
  - Replication aware
  - Transactionally safe
  - Stored procedure aware

- Dynamic Load Balancing - Optimize SQL traffic management and simplify scale out with ScaleArc's dynamic load balancing. Without any changes at the database or application level, ScaleArc lets you handle very high loads at rapid performance levels within the same data center and across multiple data centers in a way that ensures:
  - High performance
  - Transactional integrity
  - Application-level control

- Replication Awareness - With its deep intelligence, ScaleArc can route traffic based on a particular application's consistency requirements and lets you effortlessly set and enforce parameters for tolerable replication lag – all the way down to one second. Supporting Microsoft SQL Server, ScaleArc's replication monitoring engine includes:
  - Replication type and role detection
  - Replication-aware load balancing
  - Replication-aware failover

Performance

- Connection Pooling and Multiplexing - Maximize system resources and save time with ScaleArc's connection pooling and multiplexing. Because application servers don't peak together, numerous connections can site idle. ScaleArc lets you deliver only those connections required for active concurrent queries and multiplex the rest to:
  - Reduce the numbers of client connections
  - Limit the number of connections to the server
  - Make database access faster

- Query Caching - Deliver user content 6x to 60x faster by caching responses for simple and frequently repeated queries as well as for more complex queries with ScaleArc's in-memory cache. ScaleArc helps you identify queries for caching, letting you determine what to cache and for how long. With ScaleArc, you can cache:
  - Transparently and without application changes
  - In a transactionally safe manner

Analytics

- Real-time Analytics - Improve database operations with ScaleArc's comprehensive and real-time analytics.

Easily identify problematic queries or frequent queries for caching, troubleshoot performance, and assess the impact of interactional application behaviors with SQL analytics that are:

- Real-time with no sampling
- Continuously available
- Agentless

- Centralized Logging - Increase operational efficiencies and speed troubleshooting with an aggregate view of all your SQL traffic running across multiple database servers in near-real time with ScaleArc's centralized analytics. Operating as a single endpoint, ScaleArc eliminates the need to connect logs for each database, each application, and each server, delivering centralized logging that is:
  - Out of band
  - Unlimited

- Historical Stats and Forensics - Quickly go back in time, easily retrieve historical data, and get the insight you need for faster problem identification, analysis, and troubleshooting with ScaleArc's historical stats and forensics. Like a flight recorder or "black box," ScaleArc logs all essential data, so you can go back at any point in time and see:
  - How much cache is being utilized
  - How many server connections are being utilized

Chapter 2: Getting Started with ScaleArc

2.1 Upgrading from Previous Versions of ScaleArc

For information on upgrading to ScaleArc v3.8.4 from a previous version of ScaleArc, please refer to the online KB Article specifically created to assist with the planning and upgrade process.

There may be special circumstances or procedures necessary for a successful transition from an older version to the current version of ScaleArc.

Go to: **https://support.scalearc.com** and enter **`Upgrading ScaleArc`** into the search bar.

2.2 Installing ScaleArc

Getting started with ScaleArc is fairly straight forward. If you have the resource information and environment, installation of ScaleArc can be achieved in less than 30 minutes. Follow these steps to get up and running with ScaleArc:

1. Pre-requisites. Complete the **checklist guide** that includes all the necessary environment resources.

2. Installing ScaleArc: Follow the installation instructions in the ScaleArc Deployment Guide for your specific platform:
    a. Virtual Machine
    b. Physical Machine
    c. AWS Cloud
    d. Cloud

3. Configure Network Settings: Configure the basic parameters that define the **network properties** and access to ScaleArc.

4. Add your License: Apply a **license** to ScaleArc.

5. Setup High Availability (Optional): Configure **high availability** service by using two ScaleArc instances setup in a HA pair.

6. Setup a Cluster: **Create a cluster** to manage SQL Traffic.

Chapter 3: Configuring Network Settings

Welcome to the ScaleArc setup wizard! ScaleArc is installed transparently in the network between the applications and the database servers. Follow the steps below to setup networking and access credentials for this appliance.

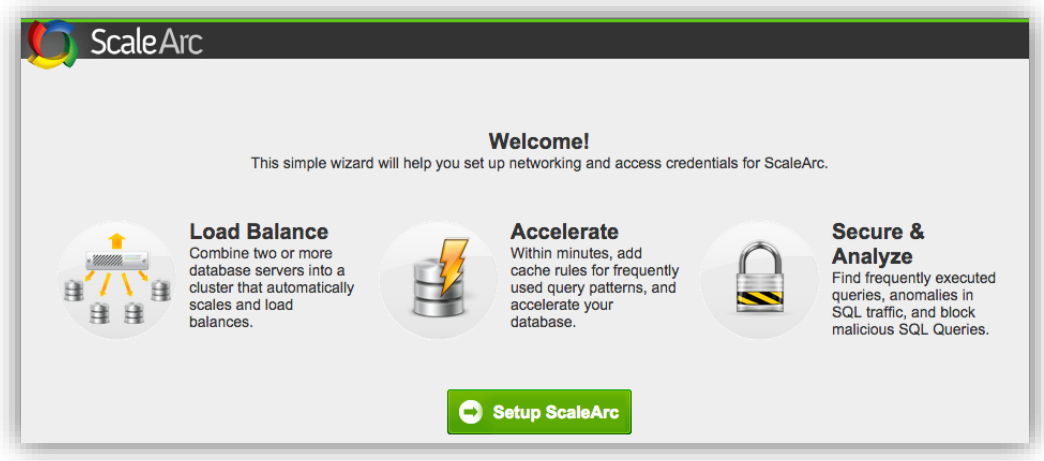


Fig 3.1

To perform initial configuration, click the **`Setup ScaleArc`** button as shown in Fig 3.1. The initial configuration screen, Fig 3.2 loads.

3.1 Network Setting

Configure the network settings used to manage the ScaleArc appliance.

Note: The basic network settings defined during the install process will be required before connecting to the UI. For more information refer to the ScaleArc for SQL Server 3.8.4 Installation Guide.

Hostname	Logical name used to manage the ScaleArc appliance. Hostname is used to identify ScaleArc software with the network.
Default Gateway	IP Address of the default gateway
Primary / Secondary DNS	IP Address for the Primary and Secondary DNS. ScaleArc uses these settings to check for software updates.
Time Zone	The time zone for the system. The default is GMT
Time Server	The NTP Server used by ScaleArc
Administrator Username	Username for ScaleArc's main administrative user, which is used for subsequent logins to this web console

Administrator Password	Password for ScaleArc's main administrative user, which is used for subsequent logins to this web console
Confirm Password	Repeat the password previously entered

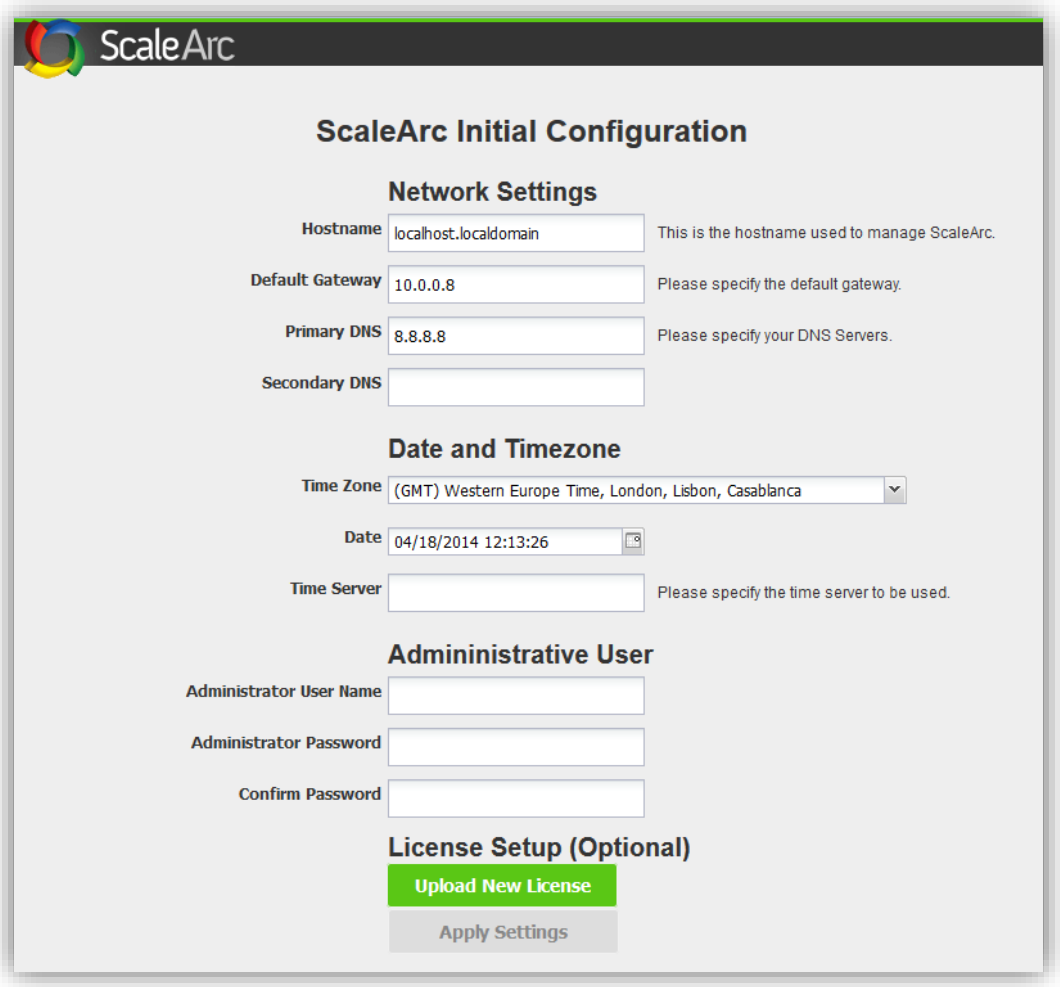


Fig 3.2

Upload New License – Click here to upload the ScaleArc license. Refer to the **ScaleArc License** chapter of this manual for more information on the different types of licenses. Wait for the appliance to upload the license and confirm it is working. This is indicated by the text on the Upload New License button turning from white to gray.

Click the **`Apply Settings`** button. The appliance loads an "Applying Configuration" dialog, then displays a "Settings are saved successfully" dialog.

Click **OK** on this dialog and then login to the web console using the administrator user name and password set earlier. Once logged in to the web console, one may clear existing events, if any. The process for clearing events is documented in the next section.

3.2 Clearing events

At this point of the ScaleArc installation, there may be some event notifications visible on the ScaleArc control panel. Click on the flashing menu bar indicator as shown in Fig 3.3:

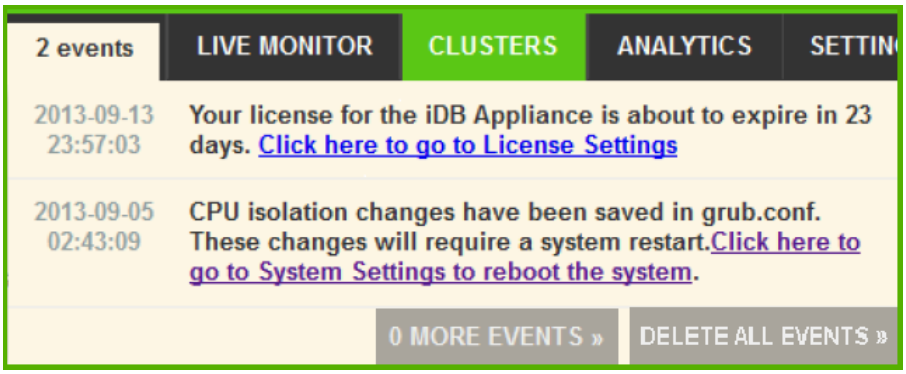


Fig 3.3

The "license…is about to expire" event is automatically posted whenever the license is scheduled to expire in 30 or less days.[1] This is a continuous event: one may clear it, but the event reappears every few minutes.

The "changes will require a system restart" event may appear after the initial installation. If this event appears, clear it by performing the following steps:

1. Click the **`0 More Events`** button on the bottom right as seen in Fig 3.3 and navigate to the Events List.
2. Click the remove icon adjacent to this event. Confirm the event's deletion.
3. Click the close icon in the Events List.

To clear all events click on the **`Delete All Events`** button.

[1] Please note all free trial licenses are valid only for 30 days.

Chapter 4: ScaleArc License

The size of a ScaleArc appliance is calculated based on the number of active cores on the database servers serving current traffic volume. In the following discussion, we'll look at each of the components required in a ScaleArc appliance and how to size them properly.

Step 1: Determine ScaleArc License Size

We determine the number of cores required in a ScaleArc appliance by examining the number of cores in the database servers that the ScaleArc appliance will be serving. Table 1 provides a summary of how the size of a ScaleArc license is calculated.

Table 1: Determining ScaleArc License Size

100% Active Database Cores Under Management	Up to 32	Up to 64	Up to 80	Up to 96	Up to 128	Up to 160	Up to 192
ScaleArc License Size	4 core	8 core	10 core	12 core	16 core	20 core	24 core

1. Count the cores in the database servers that are being used for processing live READ or WRITE database traffic.
2. Do not count passive nodes, inactive slaves, or idle backup replicas.
3. Determine what percentage of activity those cores consume at peak.

If you intentionally overprovision your database servers, include the overprovisioned cores at the desired level of activity in your count.

To calculate the number of cores required for the ScaleArc appliance, divide the number of cores by 8, and then multiply the result by the percentage of activity in those cores at peak.

ScaleArc requires a minimum of four cores in each appliance. The calculated size of a ScaleArc license must be rounded up to 4 if you calculated fewer than 4. In addition, it must be rounded up to the next even number if the calculated size of a ScaleArc license comes out to an odd number.

This value represents the size of the ScaleArc license.

Example:
If you have two active database servers, each with 48 cores, and their peak level of activity is 75%, that's 96 cores, divided by 8, which is 12. 12 times 75% is 9, and since we round up to even numbers, you would need a 10-core ScaleArc license.

Step 2: Cores in the ScaleArc Appliance

The ScaleArc appliance needs additional CPU resources for the OS and other overhead. Table 2 gives an overview of the processor core requirements in a ScaleArc appliance, depending largely on the CPU architectures in use.

Table 2: Determining the Number of Processor Cores Required in the Scale Arc Appliance

ScaleArc License Size	4 core	8 core	10 core	12 core	16 core	20 core	24 core
Intel Xeon Physical Cores (no Hyperthreading)	6	12	12	16	20	24	32
Hyperthreaded Intel Xeon Physical Cores	12	24	24	32	40	48	64
Other Cores (AMD, VM, Public Cloud)	10	16	24	24	32	40	46

The amount of overhead that's required depends on the underlying CPU architecture and the size of the ScaleArc license.

a) If you are using Intel Xeon physical CPUs, add 1 core and then add an additional core for each four cores in the ScaleArc license. If the result is an odd number, round up to the next even number. If hyperthreading is enabled in your Xeon, double the number of cores.

b) If you are using an AMD CPU or you're running on a virtual machine or in the cloud, then your ScaleArc license gives you access to an additional 50% cores. Therefore, add 1 core and an additional core plus one more core for every 4 cores in the ScaleArc license, add 50%, and round up to the next even number. These are cores that you'll use for ScaleArc even though your license size will not increase; our Fair Licensing model ensures that everyone gets the performance and capacity that they pay for.

Note that the number of ScaleArc cores may not exceed 85% of the cores in the entire ScaleArc appliance.

Step 3: Memory

Table 3: Determining Memory Requirements for the ScaleArc Appliance

ScaleArc License Size	4 core	8 core	10 core	12 core	16 core	20 core	24 core
Memory (ScaleArc Enterprise Edition)	16GB	16GB	20GB	24GB	32GB	40GB	40GB
Memory (ScaleArc Platinum Edition)	24GB	40GB	48GB	54GB	80GB	96GB	112GB

a) If you are using ScaleArc Enterprise Edition, then each core in the ScaleArc license should have 1 GB allocated to it.

b) If you are using ScaleArc Platinum Edition, then each core in the ScaleArc license should have 4 GB allocated to it.

c) The ScaleArc OS and system overhead memory require at least an additional 4 GB.

Step 4: Disk Space

Table 4: Determining Disk Space Requirements for the ScaleArc Appliance

ScaleArc License Size	4 core	8 core	10 core	12 core	16 core	20 core	24 core
OS / System Disk	100GB	100GB	100GB	100GB	100GB	100GB	100GB
Logging Disk	4TB	8TB	8TB	8TB	8TB	8TB	8TB

a) The System/OS Disk: ScaleArc requires a minimum of 100GB of space for its OS/System disk. ScaleArc recommends using solid state disks because of the increased performance.

b) Logging Disk: ScaleArc requires a minimum of 500GB of log space per core, which can be thin-provisioned on SATA storage or another slow type of disk storage. In general, ScaleArc recommends 4GB of disk space on a 4-core ScaleArc license or 8TB for more.

Logging storage requires 1GB of storage for every 2 million queries stored. More disk capacity can be added to enable retention of more historical data.

Production environments typically run with 1TB or more of disk space to retain 1-3 months of historical data. More disk space can be used to increase the number of transactions retained. If the ScaleArc runs low on logging disk space, the oldest logs get overwritten first.

Step 5: Networking

Table 5: Determining Network Requirements for the ScaleArc Appliance

ScaleArc License Size	4 core	8 core	10 core	12 core	16 core	20 core	24 core
Network	Two Bonded 1 GBE	Two Bonded 1 GBE	Two Bonded 1 GBE	Two Bonded 1 GBE	Two Bonded 10 GBE	Two Bonded 10 GBE	Two Bonded 10 GBE

ScaleArc recommends using a pair of bonded or trunked network interfaces on the appliance.

a) For a ScaleArc license of 12 cores or fewer the bonded Ethernet devices can be 1 gigabit/second.

b) For a ScaleArc license larger than 12 cores, the bonded devices should be 10 gigabits/second.

Step 6: High Availability and Redundancy

A single ScaleArc appliance can create a single point of failure. As a result, ScaleArc strongly recommends using two appliances in a redundant failover configuration for production implementations. Both appliances must be built with identical configurations in terms of the number of CPUs, memory, and disk space.

Step 7: Where to Build Your ScaleArc Appliance

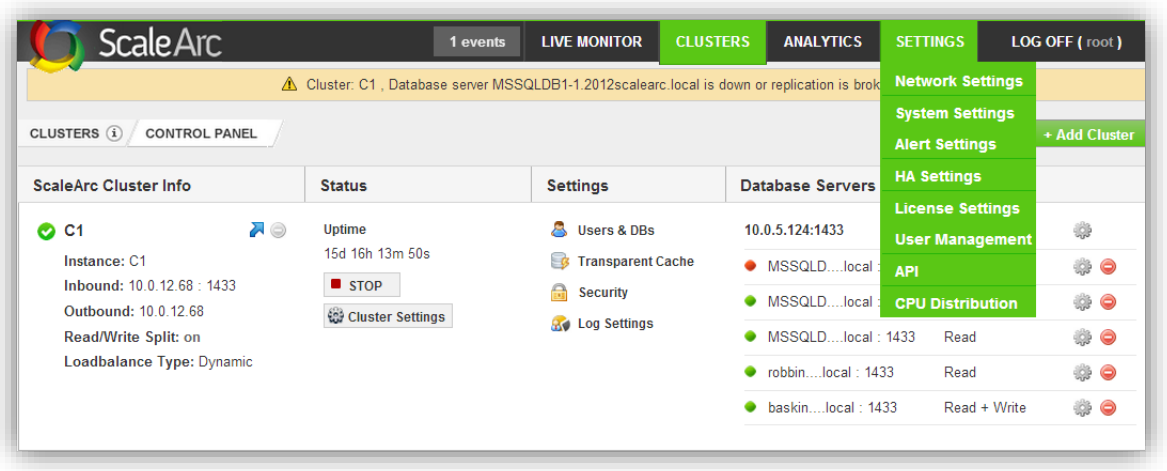
ScaleArc software can run on bare metal, on a virtual machine (HyperV, VMware, KVM, or Xen), or in the cloud. ScaleArc is supported on Amazon Web Services, Microsoft Azure, VMware Air, and many other leading public cloud providers.

Table 6: The entire ScaleArc sizing matrix

100% Active DB Cores Under Management	Up to 32	Up to 64	Up to 80	Up to 96	Up to 128	Up to 160	Up to 192
ScaleArc License Size	4 core	8 core	10 core	12 core	16 core	20 core	24 core
Intel Xeon Physical Cores (no HT)	6	12	12	16	20	24	32
Hyperthreaded Intel Xeon Physical Cores	12	24	24	32	40	48	64
Other Cores (AMD, VM, Cloud)	10	16	24	24	32	40	46
Memory (Enterprise Edition)	16GB	16GB	24GB	24GB	32GB	40GB	40GB
Memory (Platinum Edition)	24GB	40Gb	48GB	56GB	80GB	96GB	112GB
OS / System Disk	100GB	100GB	100GB	100GB	100GB	100GB	100GB
Logging Disk	4TB	8TB	8TB	8TB	8TB	8TB	8TB
Network	Two Bonded 1 GBE	Two Bonded 1 GBE	Two Bonded 1 GBE	Two Bonded 1 GBE	Two Bonded 10 GBE	Two Bonded 10 GBE	Two Bonded 10 GBE

Chapter 5: ScaleArc Configuration

This chapter describes the various ScaleArc settings and how to configure them. The System tab on the ScaleArc home page is where all the settings are accessible. The following sections in this chapter will cover each of the settings in detail.



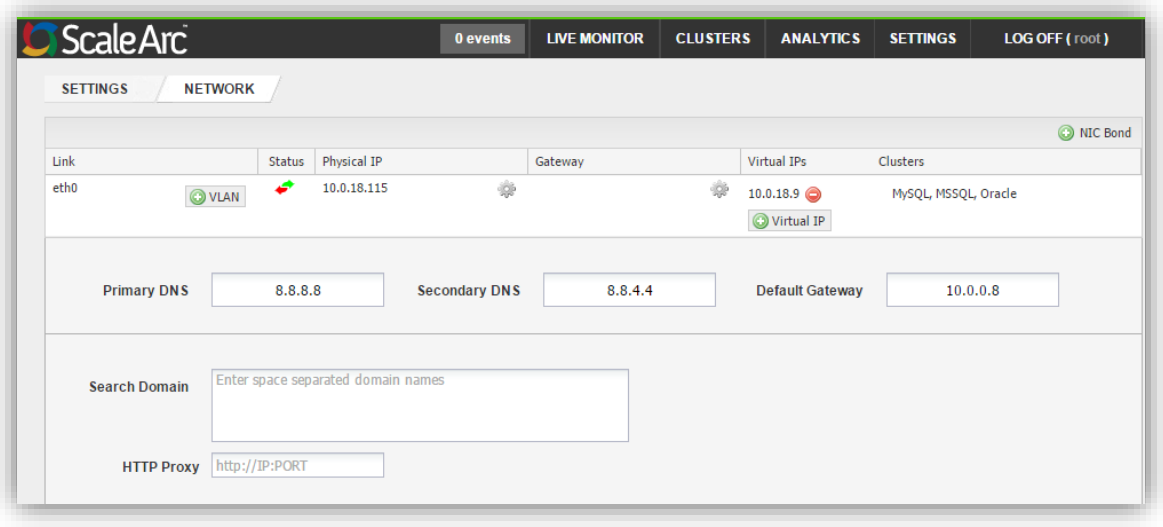
Note: This interface is only available for the management user of ScaleArc.

5.1 Network Settings

Network settings manage network interface configurations for physical as well as virtual interfaces, DNS, and default gateway. Refer to the **checklist guide** that includes all the necessary environment resources.

Note: Configuration of IP addresses should only be configured from the UI as underlying configurations within the ScaleArc framework need to be associated for proper functionality.

Note: Any field with a gear *icon next it can be selected to define or modify its value.*



The Network Settings page is divided into 6 columns. Each column in the Network Settings screen is described below:

Link – Name of the network interface on which the IP address is configured e.g. eth0.

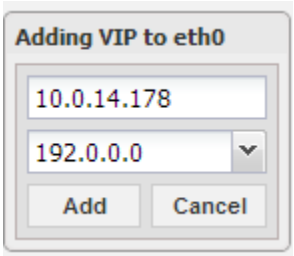
Status – Status communicates whether the link is up or down.

Physical IP - IP address for the interface. Make sure the VLAN configuration matches on the switch port.

Gateway - IP address of the network gateway.

Note:If the gateway is not defined ScaleArc will attempt to use the default gateway.

Virtual IPs - IP address assigned to the virtual interface used by clusters and high availability.



Click on **Add Virtual IP**. Enter the desired virtual IP address in the topmost entry field. Select the desired subnet mask from the drop-down control. Click on the **Add** button.

Clusters – The list of clusters configured on this interface.

Primary DNS – The IP address of the primary DNS.

Secondary DNS – The IP address of the secondary DNS.

Default Gateway – The IP address of the default gateway.

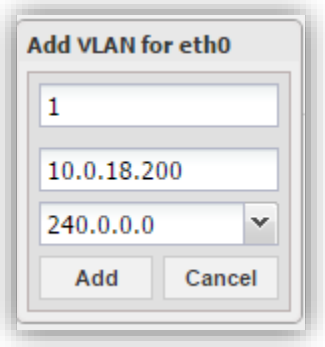
Search Domain – Search for the DNS in that particular domain.

HTTP Proxy – Configure the HTTP proxy server.

VLAN

Cick on the

option to configure VLAN.

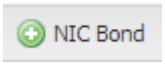


Enter the VLAN ID number you wish to to create. Assign an IP address to this VLAN. Specify the subnet and click on **Add**.

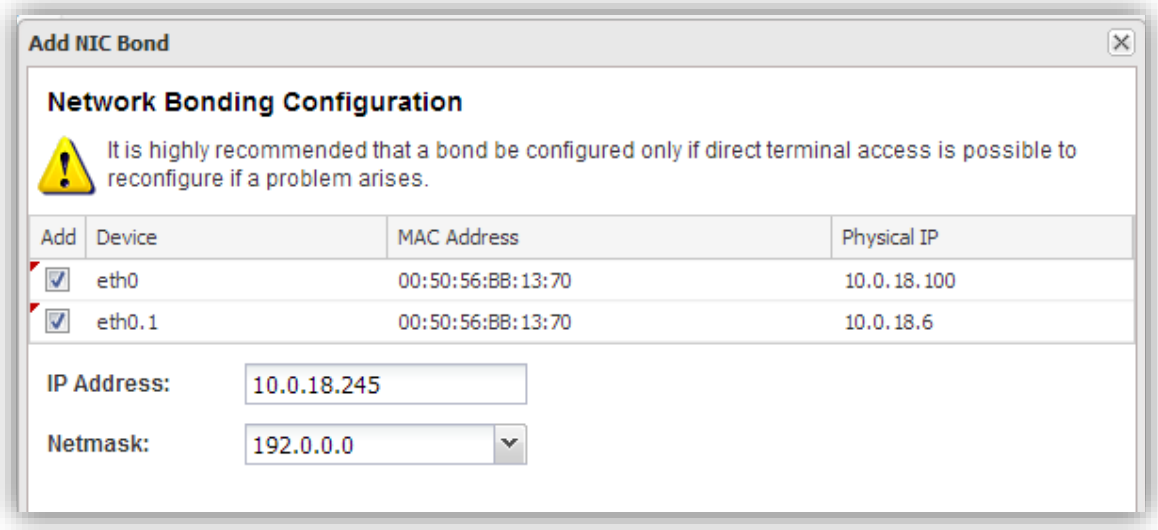
NIC Bond

NIC bonding is joining NICs (network interface cards) into one logical entity with a single IP address. There are many benefits to NIC bonding in ScaleArc such as bonding multiple bandwidths together and increasing the potential throughput of your connection.

Note: The cluster will have to be stopped before configuring the NIC bond.

Click on NIC Bond 

to configure network bonding.



Select the devices you wish to add together. Specify an IP adress you wish to assign to the NIC bond. Select the netmask and click on **Continue**.

5.2 System Settings

System Settings page is divided into 4 sections, System Status, System Config, Services, and Windows AD Setup.

5.2.1 System Status

From the System Status page you can view the following:

System Uptime – This is the time the system is working and available.

ScaleArc Software Uptime – This is the time the ScaleArc loadbalancer is up.

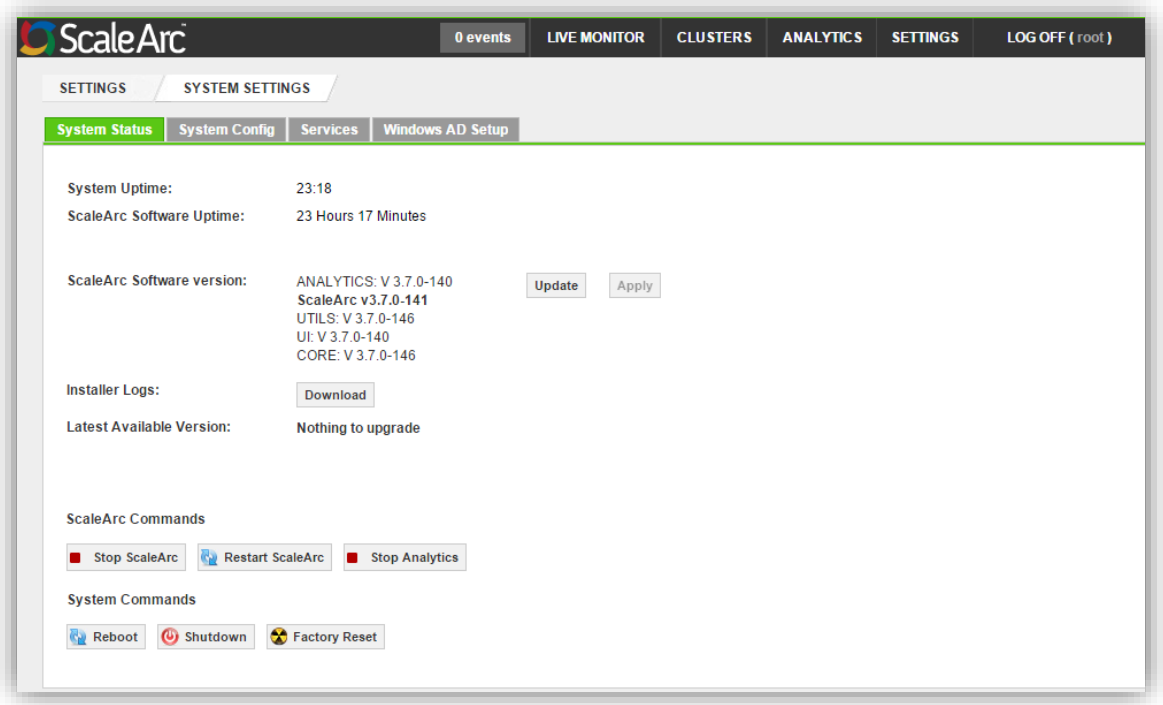
ScaleArc Software Version – ScaleArc software installed on the system. Click on **`Update Manually`** to update ScaleArc software.

Installer Logs – Logs created during ScaleArc installation can be downloaded from here.

Latest Available Version –ScaleArc will notify you if there are any software packages available for installation.

ScaleArc Commands – Start and stop the load balancer process. This applies to all core load balancing processes including cluster processes. The non-core services like UI, HA and analytics etc. will continue to execute as usual.

Start and stop the analytics process. Stopping analytics will disable analytics completely. It will stop all running analytics processes and will not process any new data.



System Commands – You can also configure options to reset, backup and restore settings. From this page you can reboot or shutdown the system and perform factory reset to reset the system back to its factory settings.

Warning: Factory reset will remove the entire system configuration. Select this option only if you want to bring the system back to its original state.

5.2.2 System Config

System configuration changes in ScaleArc can be made on this screen.

5.2.2.1 System Config

Hostname - Logical name used to manage the ScaleArc appliance. Hostname is used to identify ScaleArc software with the network.

5.2.2.2 Administrative User

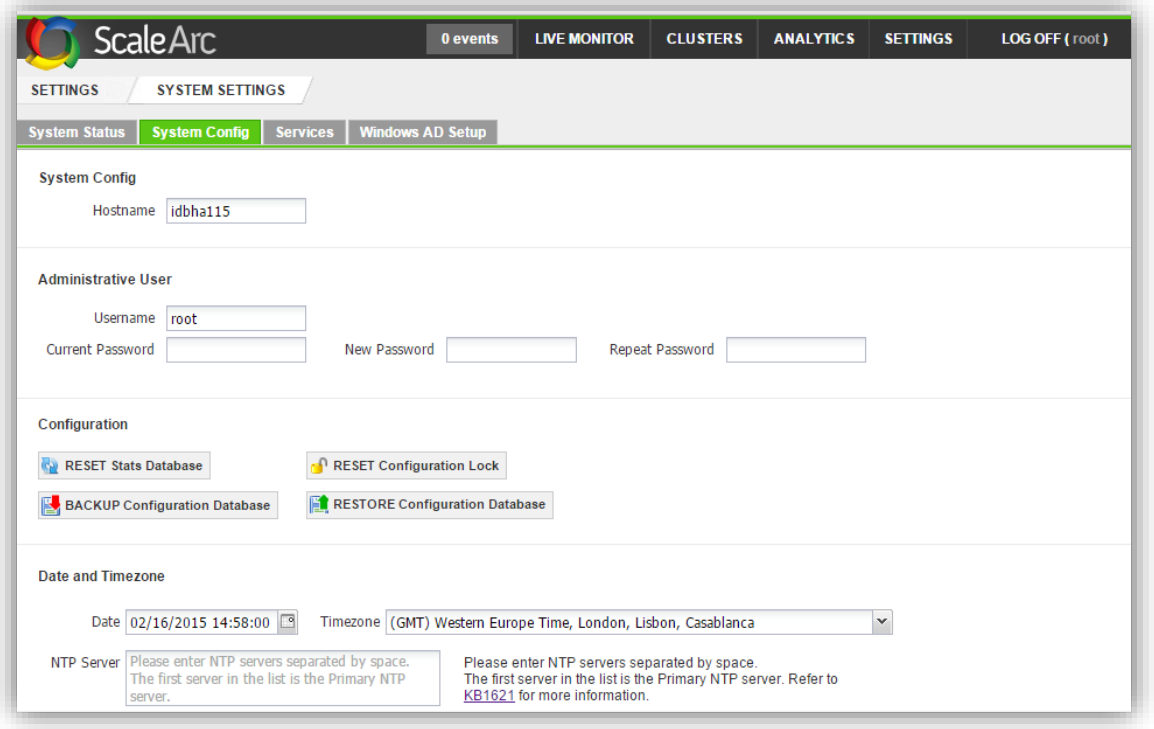
This is where the administrative user for the appliance is created and managed.

Username - The username for the administrator.

Current Password - Password for the administrator.

New Password – Create a new administrator password.

Repeat Password - Confirm the administrator password.



5.2.2.3 Configuration

Reset Stats Database –This option resets the live monitor processes so that the data on the graph is refreshed.

Warning: Reset stats database is an advance option if the live monitor graphs are stuck. There is no data loss when this operation is performed.

Reset Configuration Lock – This option should be used when you receive the following error 'Operation failed. Database is locked'. In some cases a lock of a table to the internal database is not released. This option will release the locked table for continued modification of configuration settings.

Backup Configuration Database - Create a backup copy of the current configuration database and save it to disk.

Restore Configuration Database - Restore system configuration from a file.

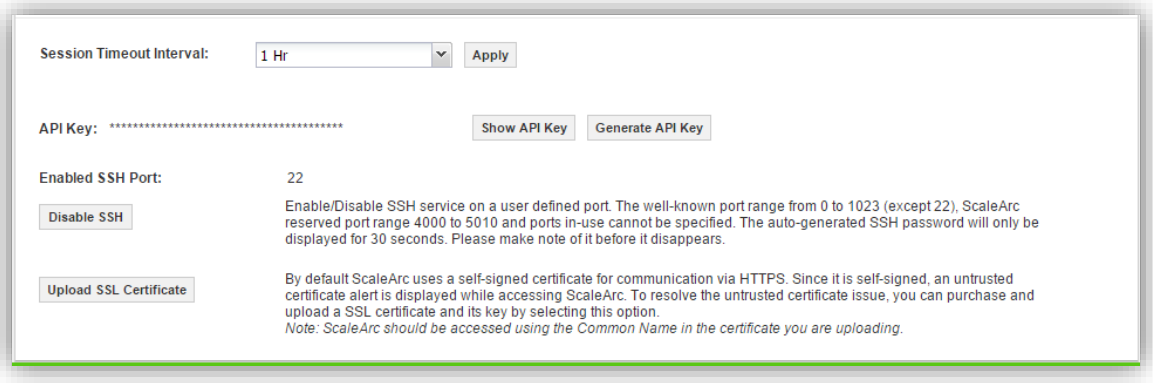
5.2.2.3 Date and Timezone

Date - Specify the system date and time in the format YYYY-MM-DD hh:mm.

Timezone – Select the time zone for the system.

NTP Server – IP address or hostname of the NTP server e.g. clock.redhat.com. You can configure multiple NTP servers (maximum of 3). Enter the servers seperated by spaces. The first server in the list should be the Primary NTP server.

5.2.2.4 Security Settings



Session Timeout Interval - A session time-out interval logs off users' inactive sessions after a specific period of time. The default value for a session time-out interval is 10 minutes. You can customize the session time-out interval for your environment and click on **Apply**.

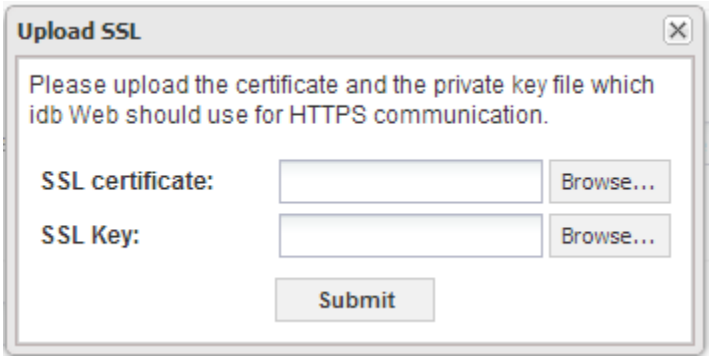
API Key - Before you can start using ScaleArc APIs, you need to identify your API key. Your API key is a unique alphanumeric identifier associated with your account and is used for universal authentication commands for all of your services. Click on **Show API Key**, enter your username and password and click on **Submit**. Your API key will be displayed. If you want to regenerate the API key click on **Generate API Key**. Enter your username and password and click on **Submit**.

Enabled SSH Port – This is the port where SSH is configured on.

Disable SSH - This option disables SSH connections on port 555. Click on **Disable SSH**, enter your administrator username and password to proceed and click on **Submit**.

Enable SSH - Click on **Enable SSH**, enter your administrator username and password and click on **Submit** to proceed. The SSH session password will be displayed for 30 seconds. Please make note of it before it disappears.

Upload SSL Certificate - By default ScaleArc uses a self-signed certificate for communication via HTTPS. Since it is self-signed, an untrusted certificate alert is displayed while accessing ScaleArc. To resolve the untrusted certificate issue, you can purchase and upload a SSL certificate and its key by selecting this option.

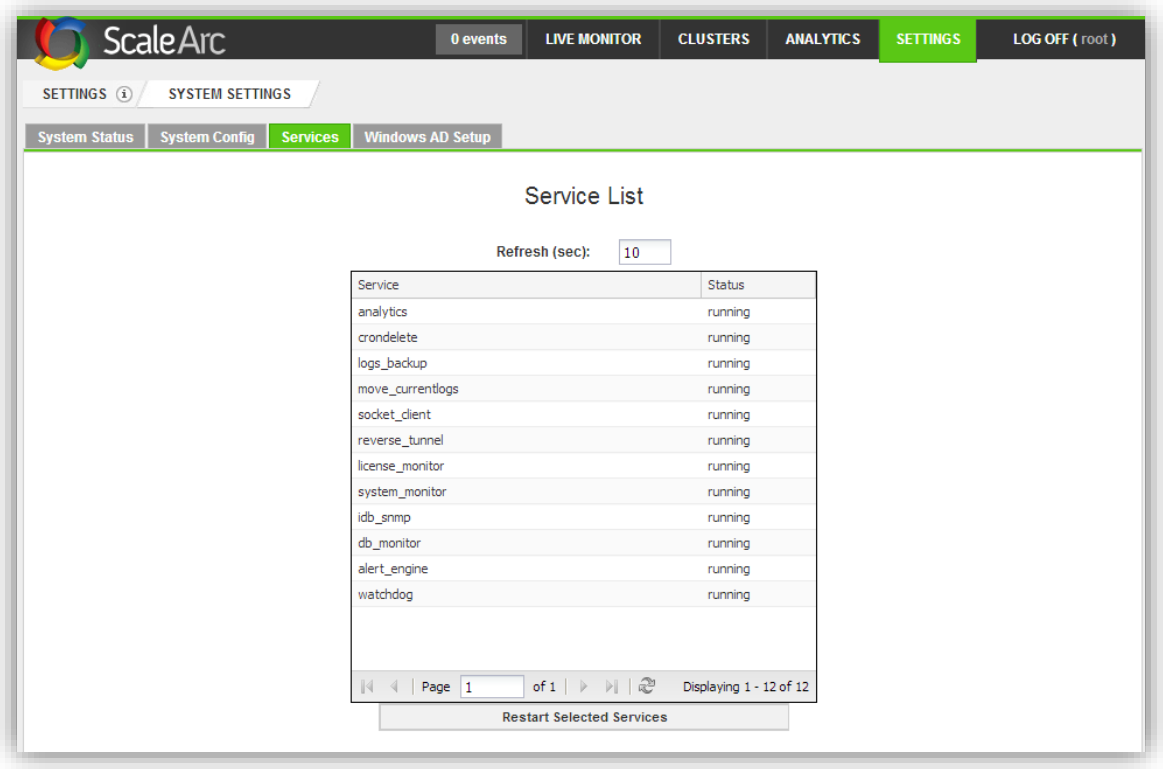


Click on **Browse** to locate and upload your SSL certificate and private key and click on **Submit**.

Note: ScaleArc should be accessed using the Common Name in the certificate you are uploading.

5.2.3 Services

The Services tab allows the user to restart running services. If a service gets stuck a database lock issue occurs. At that point you may use the Services screen to restart the service. Select the service and click on **`Restart Selected Services.`** You can change the default refresh time of 10 seconds.



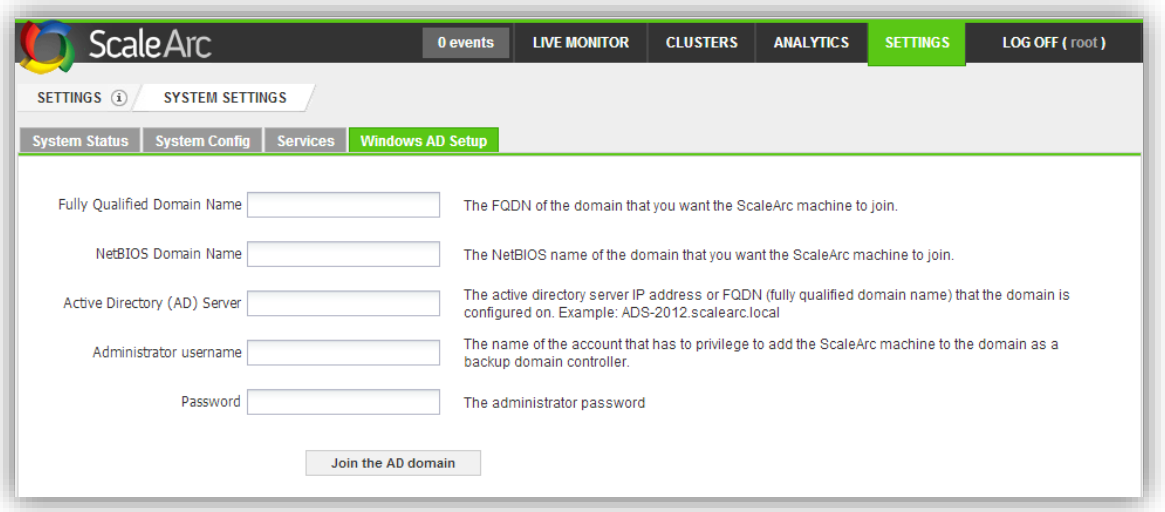
Warning: If the watchdog service is restarted, this will restart ALL other listed services.

5.2.4 Windows AD Setup

ScaleArc syncs Windows user credentials from DC by making ScaleArc a Backup Domain Controller on a Windows domain.

Before joining the domain you should have the following setup complete on ScaleArc:

1. The timezones on both the AD server and ScaleArc HA pair should be the same. This can be achieved by setting the same NTP server on all the machines.
2. Make sure the primary DNS setting on ScaleArc is correct and ScaleArc is able to access/ ping the AD server.



Fully Qualified Domain Name – Enter the FQDN of the domain that you want the ScaleArc machine to join.

NetBIOS Domain Name – The NetBIOS name of the domain that you want the ScaleArc machine to join.

Active Directory (AD) Server – The active directory server IP address or FQDN that the domain is configured on.

Administrator username – The username of the account that has the priviledge to add the ScaleArc machine to the domain as a backup domain controller.

Password – The administrator password.

5.3 Alert Settings

ScaleArc alerts help you monitor your system. ScaleArc will send an email alert when the following events occur:

Failover – This alert is generated if a failover (manual or auto-failover) is triggered and the database roles change.

Cluster health – This alert is generated if a core dump occurs.

License Expiry – This alert generates emails when the license expires or its expiry date is nearing (30 days before expiration). The alerts will be sent once every day till the license is renewed.

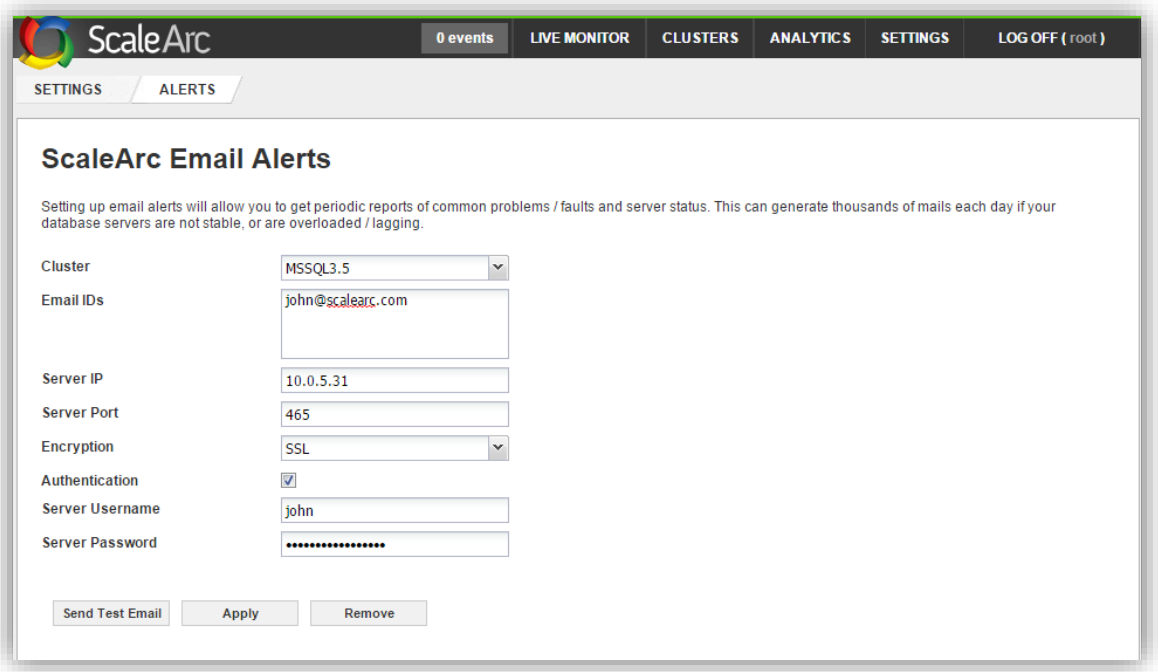
Database health – This alert is generated if the database goes down. Once the database is restored another email alert will be sent.

System health – This alert is generated if the hard drive space and/or memory goes beyond 75% of the total space. Alerts are also generated if configuration of reserved TCP Ports are changed. Setting up email alerts will allow alerts of errors, faults and server status. This can generate thousands of mails each day if your database servers are not stable, or are overloaded/lagging.

Setup an email alert by filling in the following details:

Cluster - Select the cluster from which you wish to receive alerts from.

Email IDs - List email addresses you wish to send the alerts to. Multiple email addresses are separated by a colon.



SMTP Server IP - IP address of the corporate SMTP server.

SMTP Server Port - SMTP TCP port.

Encryption – Select an option to define encryption protocol to connect to SMTP server.

Authentication – Select to authenticate outgoing SMTP mail server with SMTP username and password

SMTP Server Username - SMTP authenticated user.

SMTP Server Password - SMTP user password.

5.4 High Availability Settings

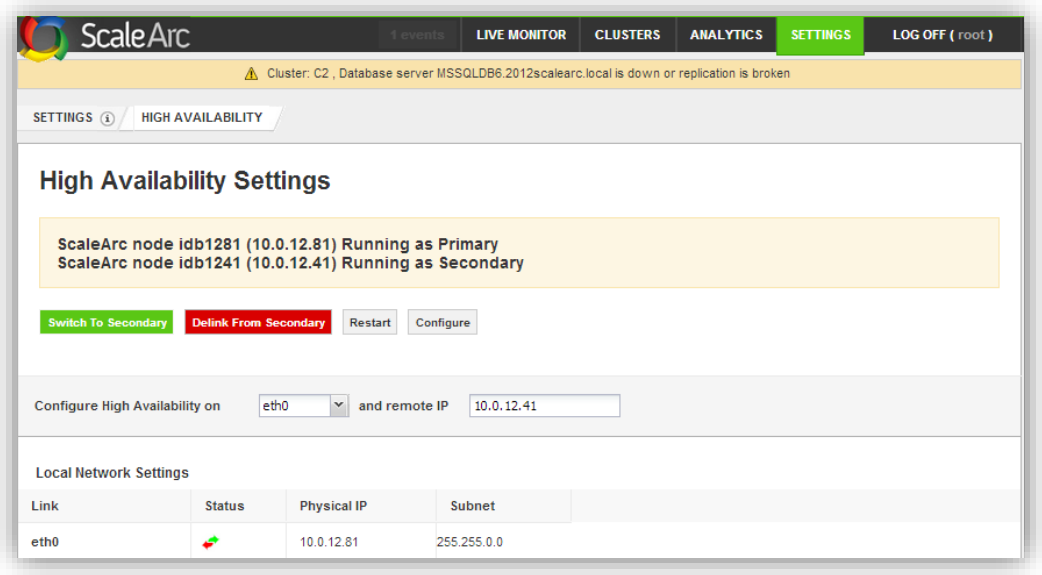
ScaleArc recommends a high availability (HA) deployment of two ScaleArc appliances, which can provide uninterrupted operation in any transaction. With one appliance configured as the primary node and the other as the secondary node, the primary node accepts connections and manages servers while the secondary node monitors the primary. If for any reason, the primary is unable to accpet connections, the secondary node takes over.

This section describes how to configure high availability for the ScaleArc machine. ScaleArc uses **Linux-HA** to setup high availability and supports active passive peer configuration with active resource sync between two nodes, virtual IP migration and heartbeat detection. The configuration page below allows an administrator to add a secondary ScaleArc machine to a standalone primary ScaleArc machine.

Note: The virtual IP migration used by the Linux HA process is not available for deployment like AWS (where IP addresses are assigned and locked to instances). For more information refer to AWS deployment guide.

Configure High Availability on – Select the interface on which you want to configure the Linux HA heartbeat service. This interface will be used for all HA communication.

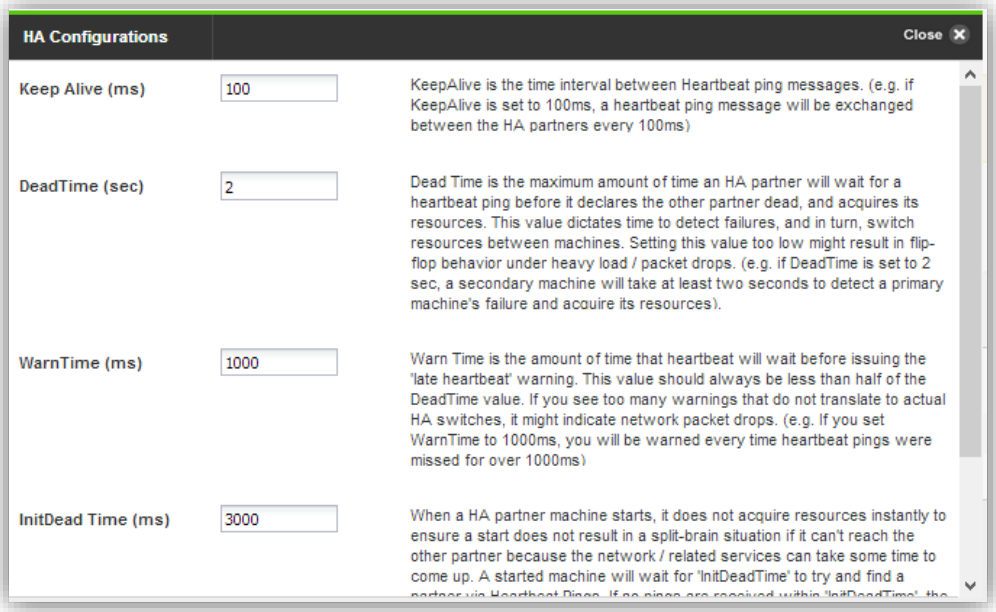
Remote IP – IP address used for HA.



Click on **`Add as Secondary`** to add the secondary ScaleArc machine.

Click on **`Restart`** to restart the HA service. This should be done only if there is any issues with ScaleArc HA service.

Click on **`Configure`** to configure advanced HA settings.



Warning: Please consult the ScaleArc support/ professional service team before making any changes. This can significantly impact ScaleArc high availability/failover.

KeepAlive – KeepAlive is the time interval between heartbeat ping messages.

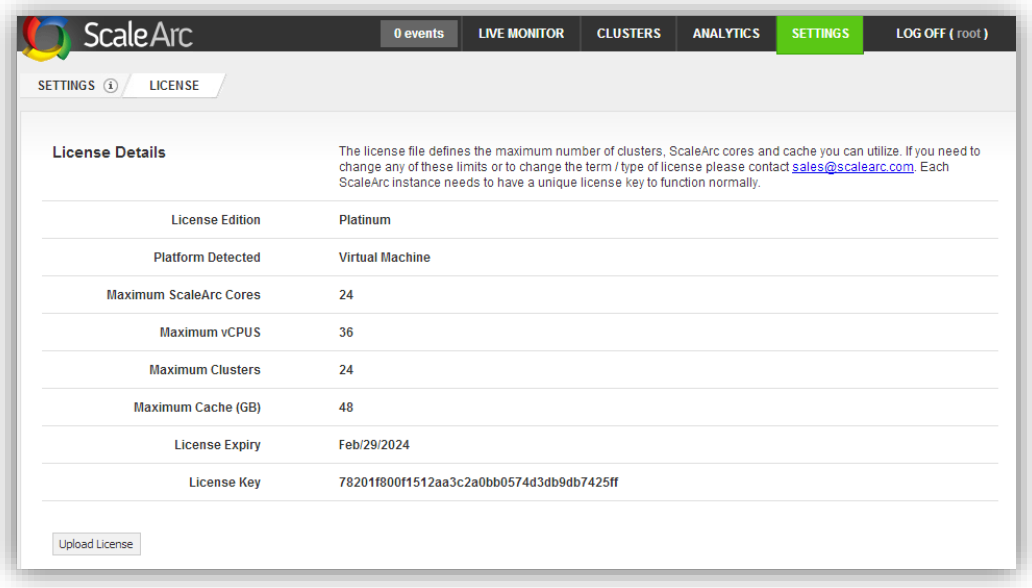
Deadtime - Deadtime is the maximum amount of time a HA partner will wait before it declares the other partner dead and acquires its resources.

Warntime -Warntime is the amount of time that heartbeat will wait before issuing the "late heartbeat" warning. This value should always be less than half of the Deadtime value.

Initdead – The Initdead parameter is used to set the time that it takes to declare a cluster node dead when heartbeat service is first started. When the heartbeat service starts it does not acquire resources instantly. This setting is to ensure that a start does not result in a split-brain situation, where a machine can't reach the other partner because the network services are taking time to come up. A started machine will wait for Initdead time to try and find a partner via heartbeat pings. If no pings are received within Initdead time, the node will become primary and acquire resources.

5.5 License Settings

The License Settings page illustrates what the current license parameters are with an option to upload new license with a file provided by ScaleArc.



License Edition - This is the software edition of the license e.g. Platinum. Refer to the **ScaleArc License** chapter for more details.

Platform Detected – Specifies the platform that the ScaleArc appliance is running on.

Maximum ScaleArc Cores: Total number of CPU cores that are allowed to be used for ScaleArc processes.

Maximum of vCPUs: Total number of virtual CPUs that are allowed to be used for ScaleArc processes. Refer to the **ScaleArc License** chapter for more details.

Maximum Clusters - Maximum number of clusters that can be added to ScaleArc independent of the database server count.

Maximum Cache - Maximum amount of memory for cache.

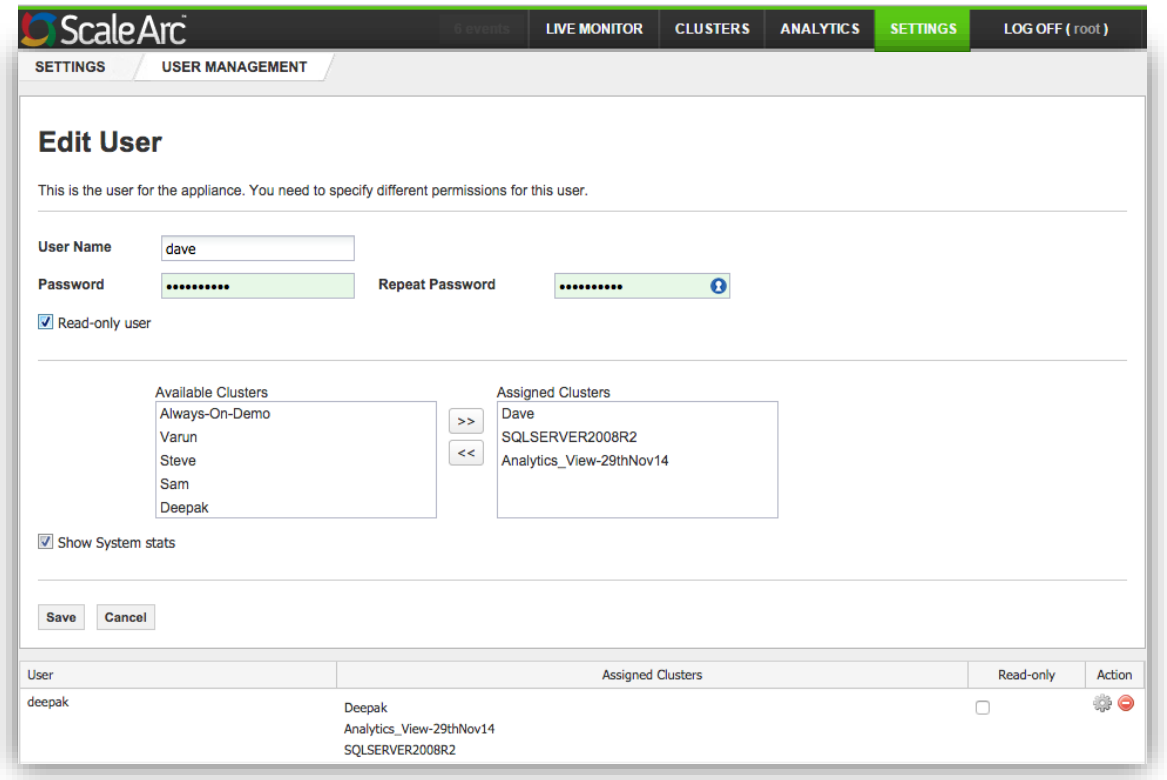
License Expiry - Date ScaleArc license expires on mmm/dd/yyyy format.

License Key – Pattern of numbers and/or letters provided to licensed users of ScaleArc.

Note: If you need to setup more clusters, or add more servers to the appliance, contact ScaleArc for a license upgrade.

5.6 User Management

The User Management screen allows us to define users and their access associated with clusters. A user must be associated with at least one cluster. Therefore you must have at least one cluster setup in ScaleArc to add additional users. See **creating a new cluster** section of this guide for more information.



Username - Login ID for the user.

Password – Password for access to the Live Monitor, Cluster, and Analytics feature for the clusters associated with user account.

Repeat Password – Type in the password again.

Read-only User – Select the check box to create a user with read-only privledges.

Available/ Assigned Clusters – Select clusters from the list of available clusters for the user being created.

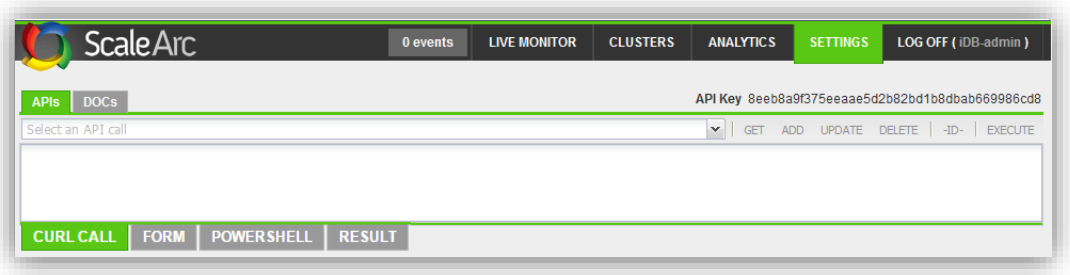
Show System Stats – Select the check box to allow the user to see the general system statics for the ScaleArc system.

To edit an user click on the gear icon under the action column. Click on the delete icon to delete a user.

5.7 API

ScaleArc can be configured using an external RESTful API. The API screen below provides information about ScaleArc API. The APIs allow a programmatic way of communication between the application(s) and the ScaleArc system to use, configure and monitor the system. The API's are based on fully RESTful HTTP service that uses all aspects of the HTTP protocol, including methods (e.g., GET, PUT, POST, or DELETE) and URIs. It uses JSON (or JavaScript Object Notation) for serializing the data in request and response formats. This screen allows you to see the various APIs, their request and response parameters and how to use them.

In the above illustration, we can see the following features of the API GUI in the top portion of the screen:



- API call drop-down, used for selecting a particular call.
- Request method pushbuttons GET, ADD, UPDATE, and DELETE.
- The -ID- field population button.
- EXECUTE button.
- API Key, to be used with all calls for this particular ScaleArc node and user.

In the bottom portion of the screen, we can see the following API GUI features:

- CURL CALL tab with text editing capabilities.
- FORM tab that generates a "fill in the blank" form for all command parameters.
- POWERSHELL tab displays API calls. These can be copied and executed verbatim on a Powershell. This feature requires Powershel version >=3.
- RESULT tab that displays the JSON result returned after executing an API command.

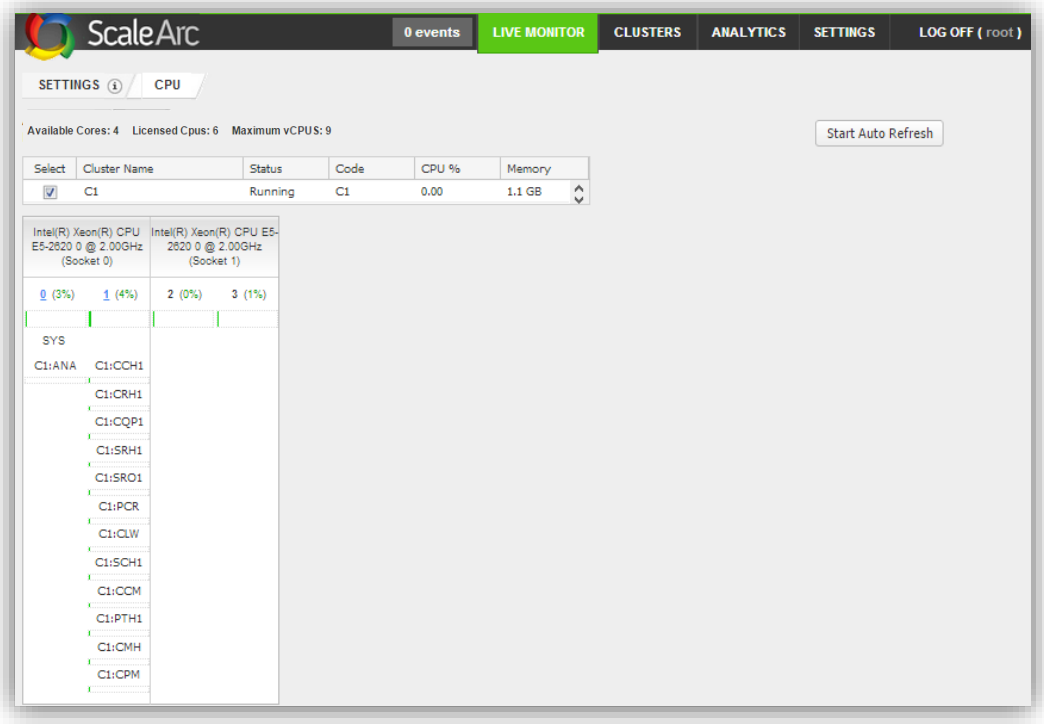
With these features, the GUI enables generation of a cURL formatted REST command by:

- Selecting the desired command from the drop-down.
- Choosing an appropriate command verb (request method).
- Identifying the specific cluster, server, or both to be used in command execution.
- Populating command parameters with a web form.
- Prototyping execution and examining command results.
- Editing the resulting command (if necessary).
- Copying the final command and pasting it into a cURL command line script.

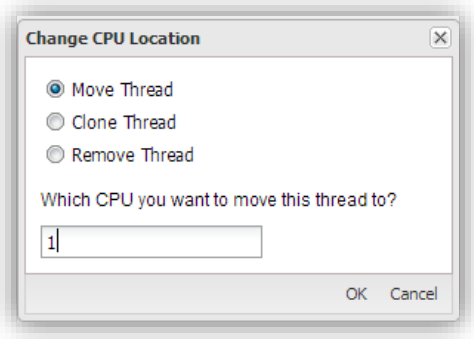
Please refer to the ScaleArc API Reference Guide for more information on the various APIs, their request and response parameters, and examples on how to use them.

5.8 CPU Distribution

This screen depicts the total number of ScaleArc processes distributed among the cores per sockets available. You can also view the status, memory utilization and total CPU utilization per cluster on a CPU on a core.



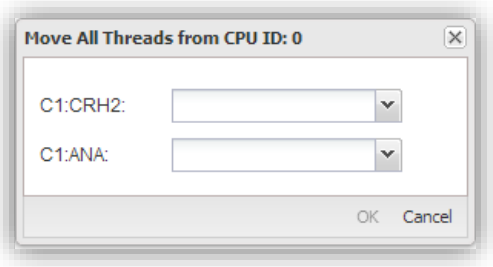
To move or clone a single thread click on the thread. You will see the following screen.



Enter the CPU you wish to move/clone this thread to and click **OK.** Select **Remove Thread** to delete the thread and click **OK.**

Note: Only cloned threads may be removed.

To move multiple threads at the same time click on the CPU ID. Select the CPU you wish to move the thread to from the drop down list and click on **OK**.



Note: Inactive threads are in red color while active threads are in blue. A CPU running a load of more than 60% utilization shown in green will become red.

As the threads are dynamic they switch between multiple states of active and inactive.

Chapter 6: Clusters

This chapter describes creation, administration and management of clusters in ScaleArc.

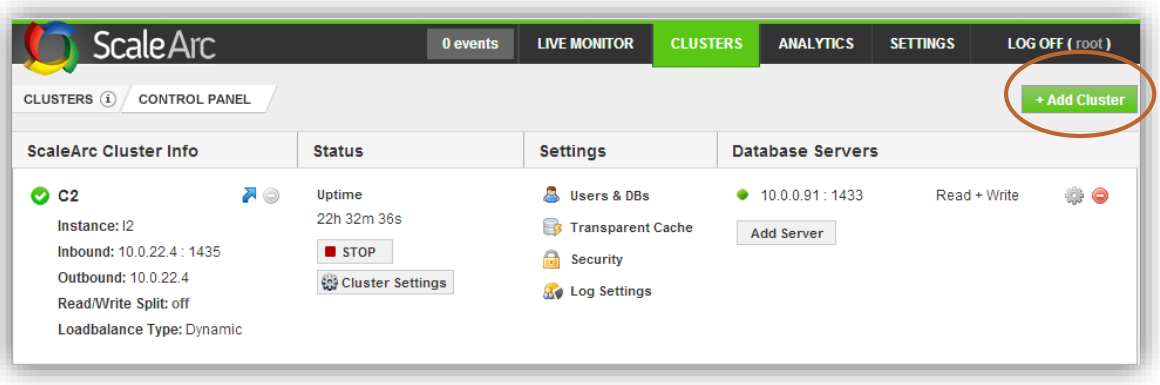
6.1 What is a Cluster?

Each ScaleArc cluster serves as a unique logical load balancer providing real-time SQL protocol analysis and monitoring, user access permissions, user transaction level permissions, connection authentication, transparent read/write split offload, SQL surge queue, cache offload rules, firewall rules, pre-cache rules, log management, analytics and the ability to manage (add, remove and modify) database servers as part of a cluster. Each of these features support a great degree of management as well. Each ScaleArc cluster is a unique user process. Stopping/ starting one cluster does not impact traffic on another cluster.

6.2 Creating Clusters

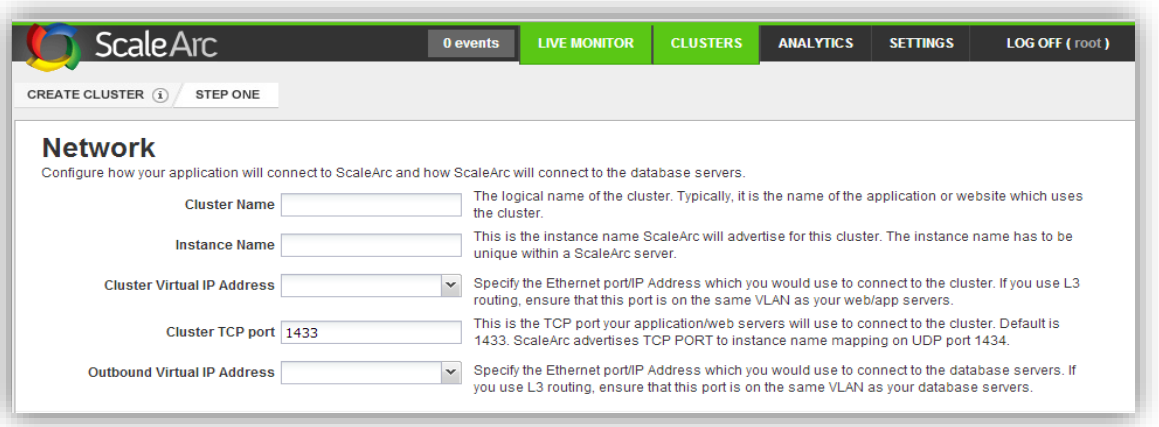
Clusters are logical load balancers and each can be configured to uniquely manage SQL traffic. While configuring each cluster in ScaleArc we will define the inbound interface and a listening port as well as the outbound interface for each database server.

After we complete initial configuration, the appliance is ready to form the initial cluster. Click on **`Add Cluster`** to begin the cluster creation process.



Network

First step is to configure your network. This is where you configure how your application will connect to the load balancer and how the load balancer will connect to the database servers.



Cluster Name - Enter a cluster name, such as Accounting or Product-Catalog. This name is descriptive and used throughout the web console.

Instance Name – This is the instance name ScaleArc will use for this cluster. The instance name must be unique within a ScaleArc server.

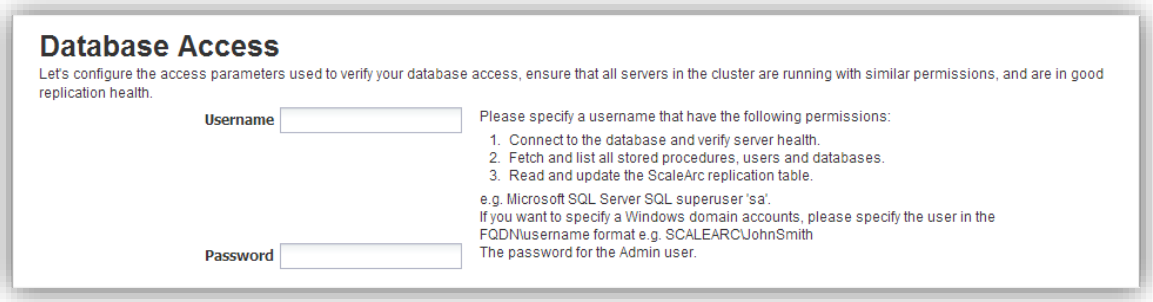
Cluster Virtual IP Address - Select a load balancer virtual IP address from the drop down list. This is the IP address that the applications will now use to connect to ScaleArc.

Cluster TCP port - Set the database TCP port, 1433 is the default.

Outbound Virtual IP Address - Select a load balancer virtual IP address from the drop down list. This is the IP address that ScaleArc will use to connect to the database servers.

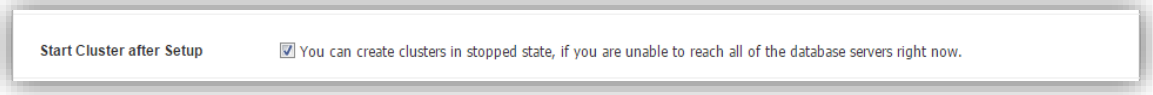
Database Access

The next step is to configure the access parameters used to verify your cluster, ensure that all servers in the cluster are running with similar permissions, and are in good replication health. ScaleArc needs to be able to browse the database server and check which user has access to which database server. This user also needs to be able to execute show slave status so that ScaleArc can monitor replication lag.



Username – Super user for the database server.

Password – Password for the administrative user.



Start Cluster after Setup – Select this option if you would like to start the cluster after setup. Alternatively, you can choose to create a cluster in stopped state if you are unable to reach all of the database server.

Cluster Settings

You can choose to copy cluster settings from another cluster from the drop down box.

SSL Settings

ScaleArc supports TDS over SSL. If the SQL Server requires SSL, check the box to enable SSL for the Cluster. You will need to provide a Server certificate and Server Key to ScaleArc. The SSL Certificate must be generated by a CA-Authority that is already approved by the application hosts for it to be accepted by the application hosts.

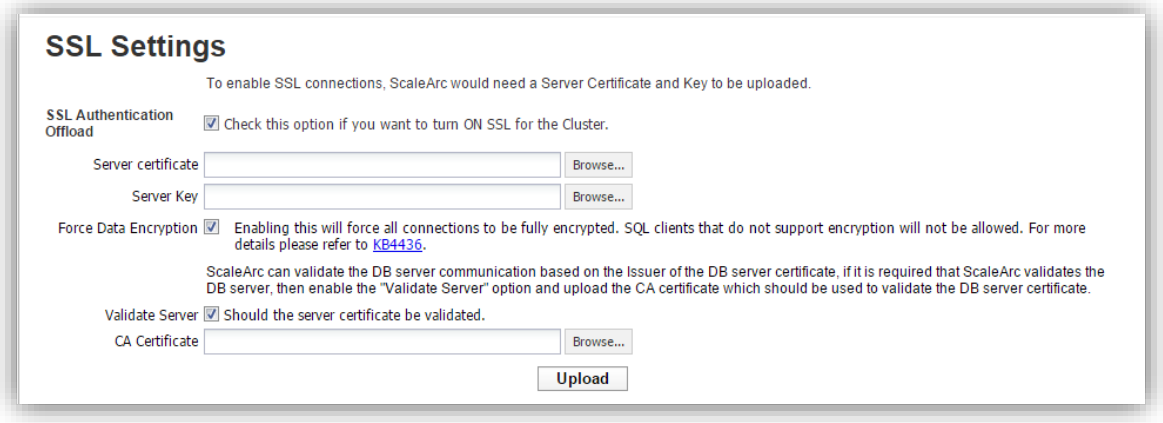
Note: Any modification to the SSL Offload option would require a cluster restart.

Enabling **`Force Encryption`** will force all connections to be fully encrypted. SQL clients that do not support encryption will not be allowed.

SSL offload allows you to establish secure communication between the client and the server via ScaleArc.

Important - as a prerequisite, if a SQL client needs to initiate full encryption with ScaleArc's SSL enabled cluster, ScaleArc inbound IP should be reverse resolvable to the hostname entry in certificate uploaded on ScaleArc cluster.

Select the **`Validate Server`** option if you require ScaleArc to validate the DB server communication and upload the appropriate CA certificate.



Servers

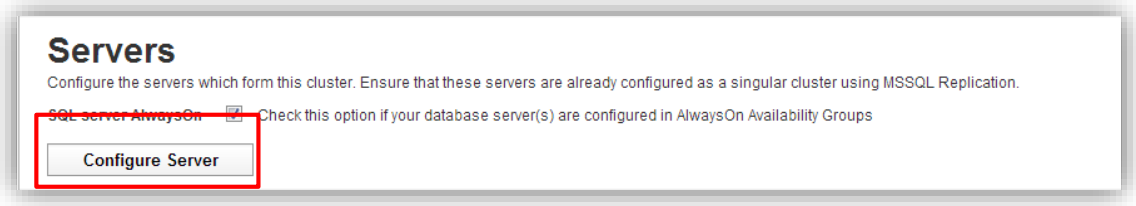
Add a database server to the cluster that's being created. At this point you can add as many servers or just one. Keep in mind, one of the benefits of ScaleArc is its ability to load balance across multiple databases. Database servers can be modified later as well. If adding more than one server you must ensure these servers are already configured as a singular cluster using **Microsoft SQL Server 2008 R2 Replication**. Please make sure the cluster outbound LAN connection IP address is allowed to connect to the database servers.

Configuring AlwaysOn Server

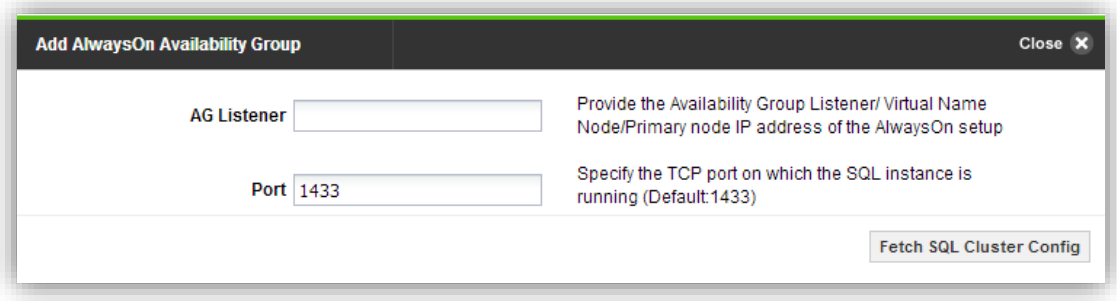
Select the check box **`Check this option if your database server(s) are configured in AlwaysOn Availability Groups`** to configure a SQL 2012 AlwaysOn server.

Note: For proper Always-On server discovery and failover operations, the 'SQL Server Browser Service' must already be running on each of the DB servers. For more information refer to the following article: **How to start and stop the SQL Server Browser Service**

Click on **`Configure Server`**.



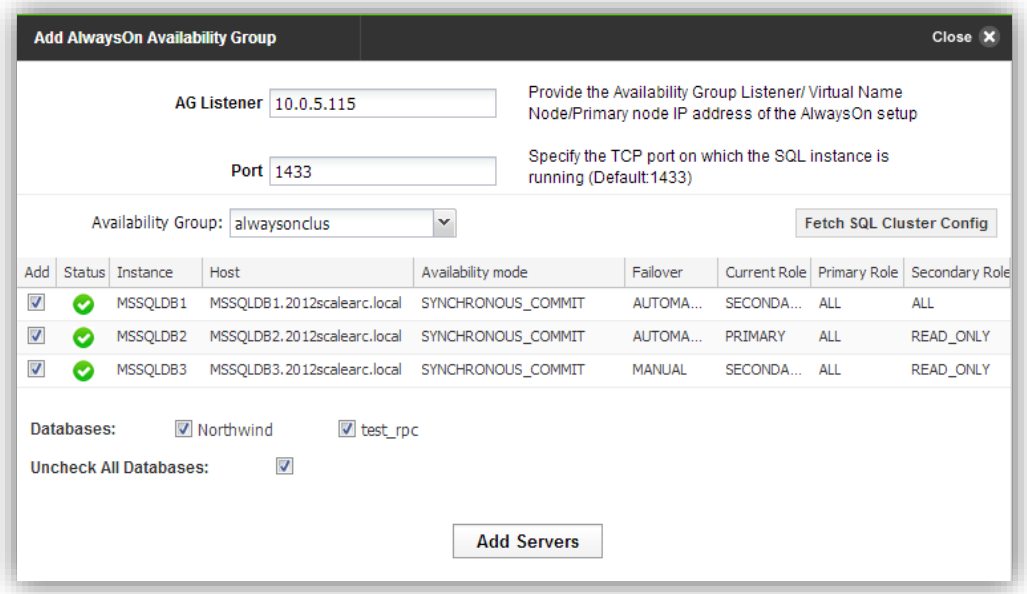
On the next page you can specify a VNN address and the port on which the VNN services are running.



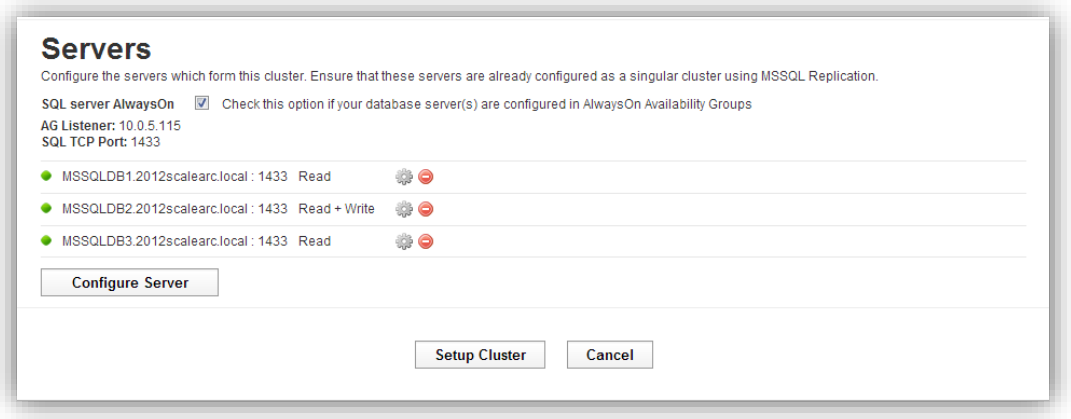
Specify the Availability Group Listener/ Virtual Name Node/ Primary Node IP address of the AlwaysON setup.

Specify the TCP port on which the SQL instance is running (default is 1433).

Click on **`Fetch SQL Cluster Config`**. This will connect to the VNN and fetch all the databases which are configured for this cluster. You can use the check boxes to select the databases you would like to add.



Click on **`Add Servers`** to add the selected servers to the AlwaysON Availability Group.

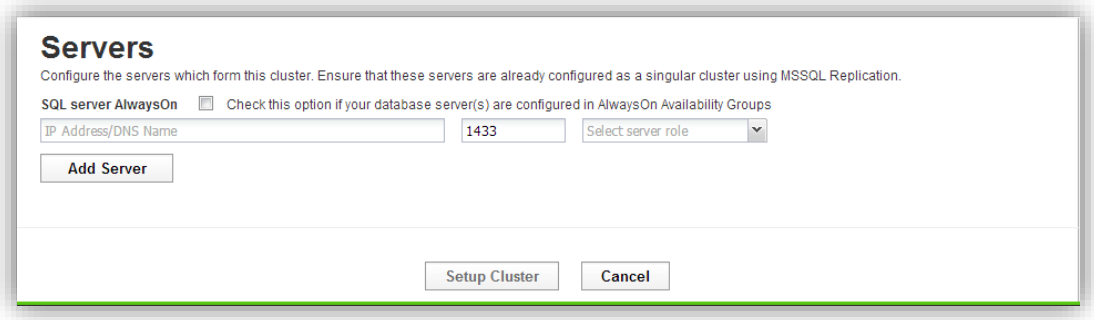


Click on **Setup Cluster** to complete the cluster creation process.

Configuring Stand-A-Lone or Transactions Replication servers

Follow the steps below to configure server settings without the AlwaysOn feature.
Add IP address/ DNS name of the server you wish to add. Select the server role from the drop down list and click on **Add Server**.
Server Modes:

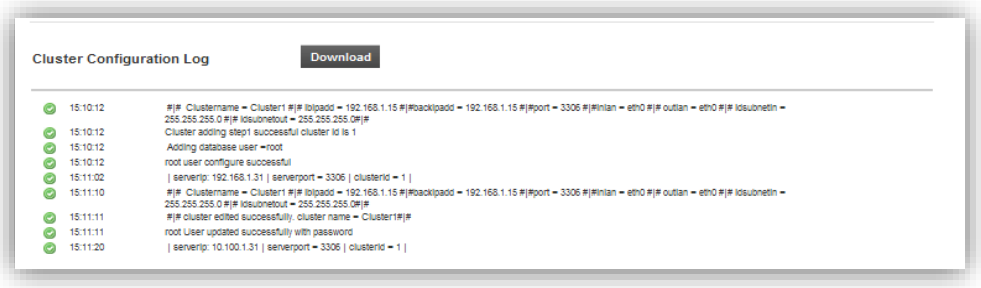
Read+Write	Setting for primary Servers. This setting tells ScaleArc that this server can take write operations.
Read	Setting for secondary or backup Servers that can take read-only transaction at run time.
Read + Standby	Setting for secondary machines that can take read or write transactions but ScaleArc will only deliver read queries to this server type. In the event of the Read+Write server becoming unavailable ScaleArc will automatically promote servers with this type to be a Read+Write server and demote the original server to Read+Standby. This is most often used for transactional replication sets.
Standby, No Traffic	Warm standby server for ScaleArc. This server cannot or should not take transactions at run time unless the Read+Write Server becomes unavailable, at which time ScaleArc will demote the Read+Write server and promote the Standby Server to be Read+Write. Often used for other replication types for SQL Server like log shipping.



Click on **Setup Cluster** to complete the cluster creation process.

Cluster Configuration Log

Click **Download** to download a list of clusters and their settings.



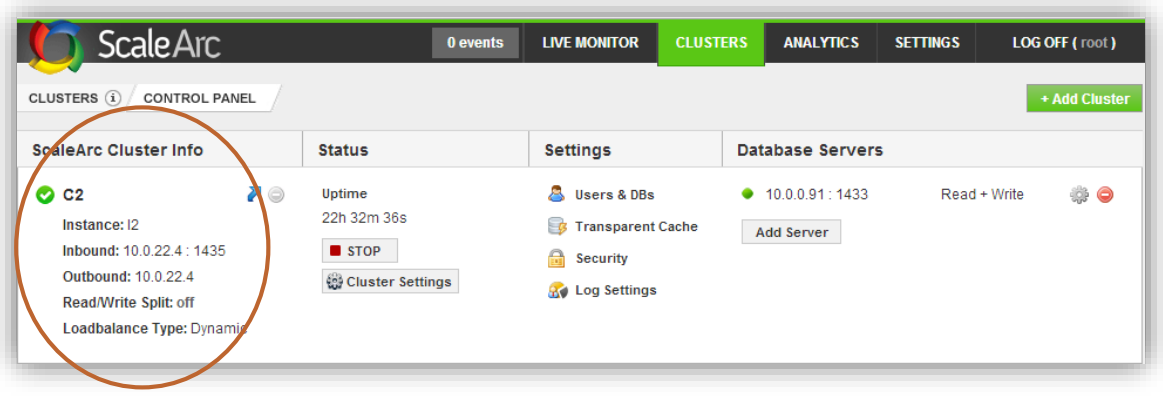
Click **Finish**.

Your cluster has been successfully created and configured. At this point you can add more users and configure their database access or continue to the cluster settings page.

6.3 Managing Clusters

The clusters screen is divided into 4 parts ScaleArc Cluster Info, Status, Settings and Database Servers.

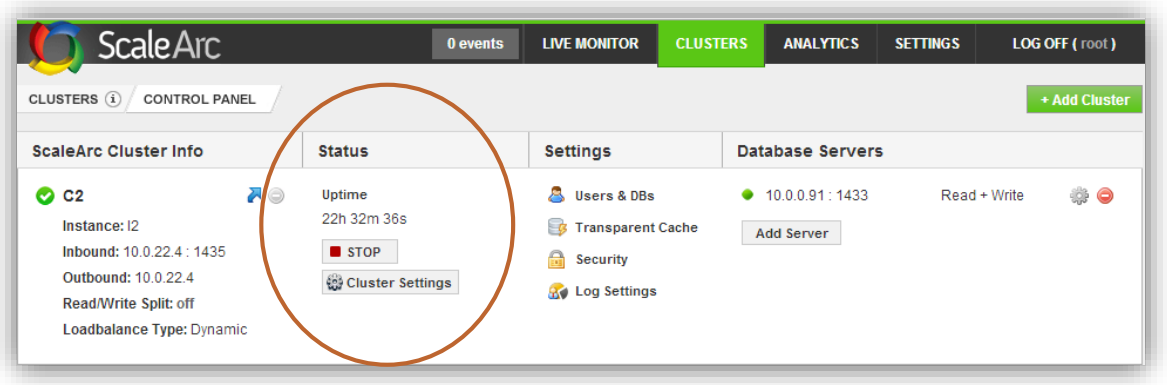
6.3.1 ScaleArc Cluster Info



The ScaleArc Cluster Info section lists out the properties of the cluster. In this section you will see the cluster instance type, inbound/ outbound IP address, whether read/ write split is ON/OFF, and the load balance type.

6.3.2 Status

Cluster status reports on the overall uptime of the cluster in weeks, days, hours, minutes and seconds. It provides start and stop management for the load balance service on the subsystem.



Click on **`Cluster Settings`** to modify or enhance the configuration of the clusters. These include the same options presented when we first configured them with the cluster creation wizard.

6.3.2.1 Cluster Settings

Cluster Name – A name which uniquely identifies the cluster within the interface.

Instance Name – This is the instance name ScaleArc will use for this cluster. The instance name must be unique within a ScaleArc server.

Read/Write Split - ON/OFF – Enabled when multiple database servers are defined in the cluster. ScaleArc distinguishes read from write statements and routes traffic accordingly. Read traffic is load balanced across multiple servers leaving writes to be processed by the master server, defined in ScaleArc as Read/Write. For typical master slave environments read/write split has to be turned ON so that ScaleArc can send reads to slaves and everything else (+ some read traffic) to masters.

Cluster Analytics - Determines if analytics are running for this specific cluster.

Inbound/Outbound IP- LAN Interface, IP Address and Port settings.

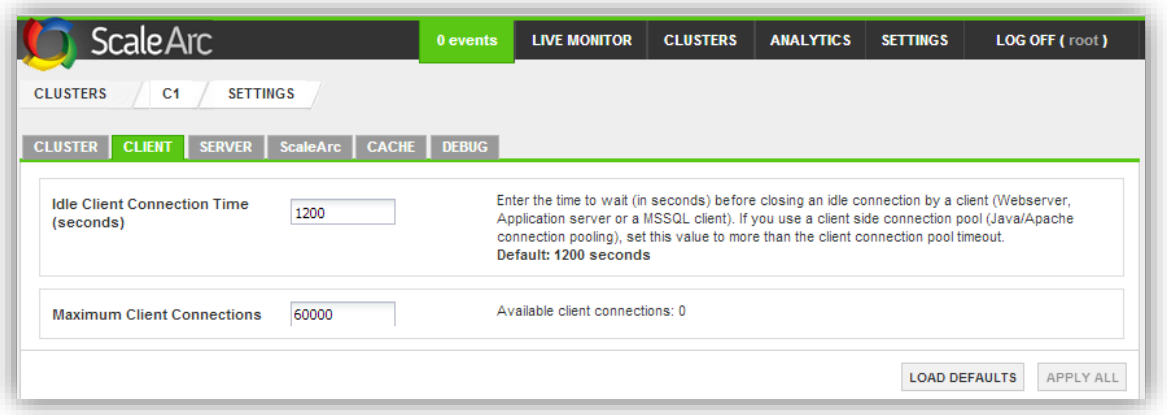
Port - Specify the TCP port which you'd like to use to connect to the cluster.

Load Balance Type (dynamic load balancing or round robin) - As such with traditional load balancing technologies, ScaleArc load balancing distributes SQL traffic across multiple database servers based on availability and configuration variables established for each cluster. Availability, replication lag, max concurrent server connection, load balancing type and time to first byte all play a critical role, which determines how traffic is distributed.

Note: ScaleArc is a dynamic load balancer and adjusts its server rotation on fly based on the metrics mentioned above.

6.3.2.2 Client Settings

Warning: Do not make changes to the Client Settings if you are not sure about what you are doing. Contact ScaleArc customer support with any questions you may have.



Idle Client Connection Time (seconds) - Enter the time to wait before closing an idle connection by a client (web / application server / SQL Server Client). If you use a client side connection pool (Java/Apache connection pooling), set this value to more than the client connection pool timeout. The default value 1200 seconds.

Maximum Client Connections – The maximum number of client connections allowed for the cluster. This is now kept at a default maximum of 60000. The depth of the surge queue is also limited by this number.

6.3.2.3 Server Settings

Click on the Server Settings tab to customize the server configuration.

Server Connection Timeout (seconds)– Enter the maximum time after which the server connection will be closed if ScaleArc could not open the TCP connection with the server. After the timeout occurs the corresponding client connection will also be closed.

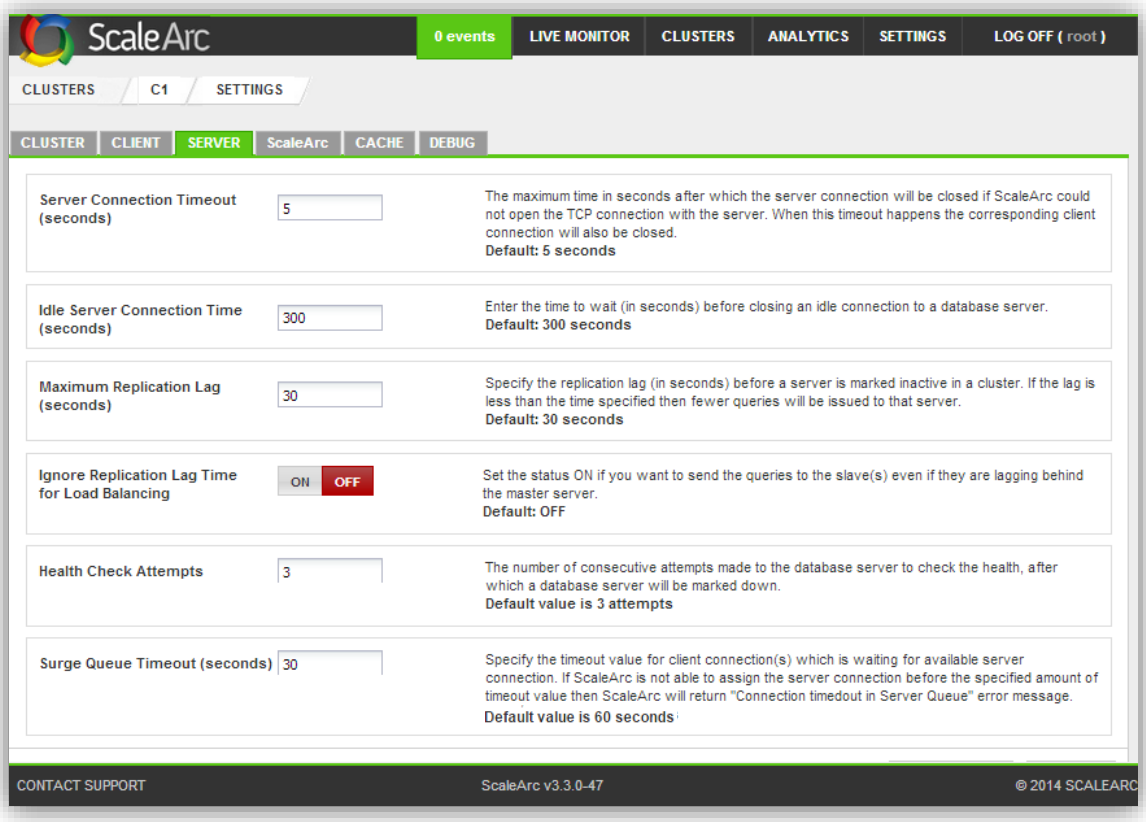
Idle Server Connection Time (seconds) - Enter the time to wait before closing an idle connection to a SQL Server. Preferably set this to more than 120 seconds. The default value is 300 seconds.

Maximum Replication Lag (seconds) - Specify the replication lag before a server is marked inactive in a cluster. If the lag is less than the time specified then fewer queries will be issued to that server (default value is 30 seconds).

Ignore Replication Lag Time for Load Balancing (ON/OFF) - Turn this status ON if you want to send the queries to the slave(s) even if they are lagging behind the master server. The default status is OFF.

Health Check Attempts - The number of consecutive attempts made to the database server to check the health, after which a database server will be marked down.
The default value is 3 attempts.

Surge Queue Timeout (seconds) –This is the timeout value in the application or the connection pool configuration which determines the maximum length of time a connection can be held in the surge queue before returning an error message.



6.3.2.4 ScaleArc Settings

Click on ScaleArc Settings tab to modify or enhance the configuration of ScaleArc.

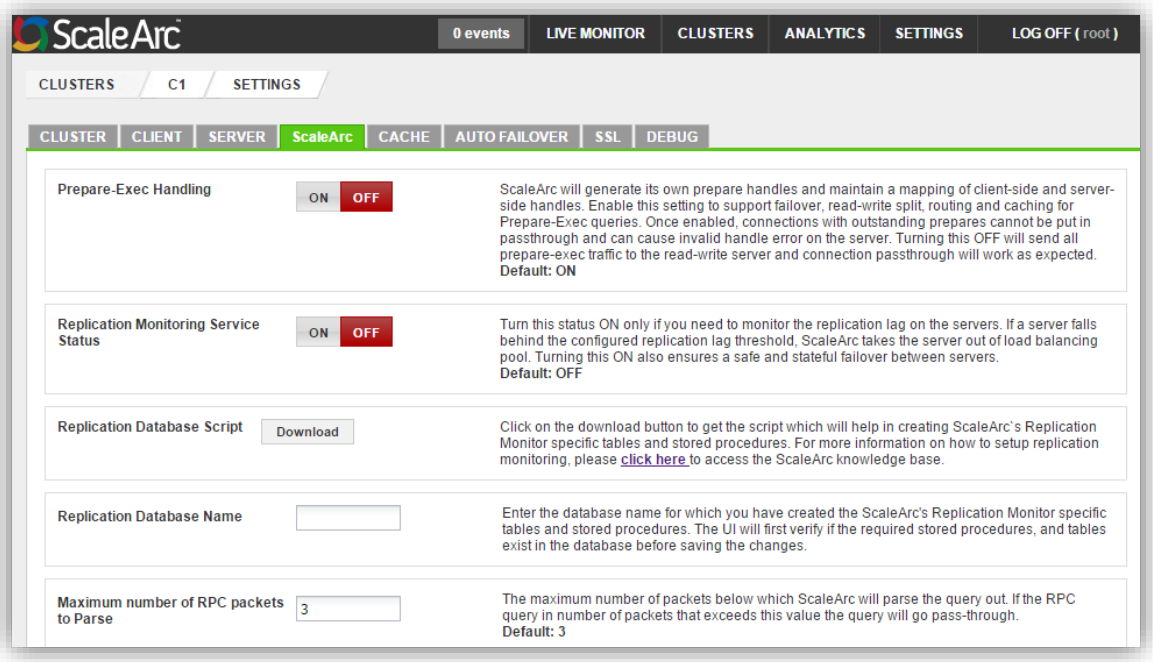
Prepare-Exec Handling – ScaleArc will generate its own prepare handles and maintain a mapping of client-side and server-side handles. Enable this setting to support failover, read-write split, routing and caching for Prepare-Exec queries. Once enabled, connections with outstanding prepares cannot be put in passthrough and can cause invalid handle error on the server. Turning this OFF will send all prepare-exec traffic to the read-write server and connection passthrough will work as expected.

Replication monitoring service status – This feature when turned ON monitors the replication lag on the servers. ScaleArc writes to a certain table on the primary server which will get replicated on the secondary servers.

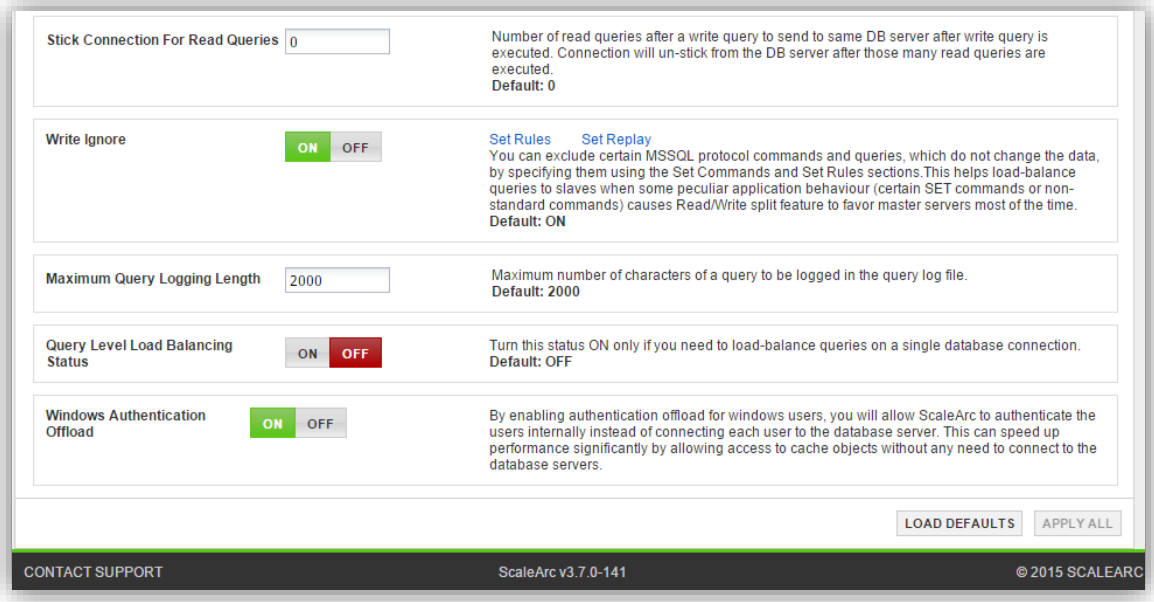
Replication Database script – Click on the download button to get the script which will help in creating ScaleArc's replication monitor specific tables and stored procedures.

Replication Database Name - Enter the database name for which you have created the ScaleArc's Replication Monitor specific tables and stored procedures. The UI will first verify if the required stored procedures and tables exist in the database before saving the changes.

For more information regarding the Replication Monitoring service in ScaleArc please refer to the **knowledge base article**.



Maximum number of RPC packets to parse – The maximum number of packets below which ScaleArc will parse the query out. If the RPC query in number of packets that exceeds this value the query will go pass-through. (Default value is 3)



Stick Connection for Read Queries – Specificy the number of read queries after a write query to send to same DB server after write query is executed. Connection will un-stick from the DB server after those many read queries are executed.

Write Ignore (ON/OFF) - You can exclude certain MS SQL Server protocol commands and queries, which do not change the data, by specifying these using the Set Rules section. This helps load balance queries to slaves when some connection pooled application behavior (certain SET commands or non-standard commands) causes read/write split feature to favor

master servers most of the time. For connection pooled apps like Java/Ruby on rails, setting write ignore and set ignore rules along with turning on query level load balancing is required to achieve read/write split. Adding a set option like "SET names utf8" would let ScaleArc know that this set rules is okay to be sent to a read server. Consider putting the same SET rules into SET replay as well to retain connection level fidelity. When ScaleArc switches a connection from the read server to a write server, it will not replay the SET commands, to achieve this one has to put all of the necessary commands in SET replay. If you have 2 applications one connection pooled and the other non-connection pooled one- use two different ScaleArc clusters so that different advanced settings can be applied for optimal application compatibility and performance

Set Rules – Click on this link to exclude certain MS SQL Queries from being detected as write queries.

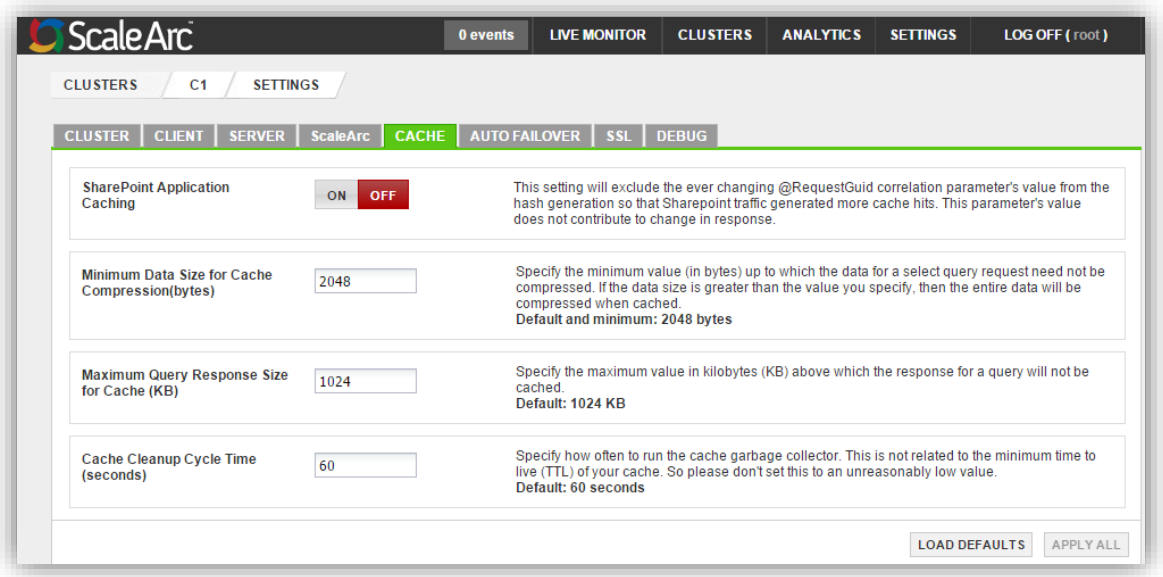
Set Replay – Click on this link to specify a SQL command or statement which will get executed on every new connection which ScaleArc uses after switching the connection from one database server to other database server. The switch can happen due to following reasons, query routing, Read/Write split, query level load balancing.

Maximum Query Logging Length – Maximum numbers of characters of a query to be logged in the query log file. (Default value is 2000)

Query Level Load Balancing Status – Turn this status ON only if you need to load balance queries on a single database connection.

Windows Authentication Offload – Turn this feature ON to enable ScaleArc to authenticate Windows users internally instead of connecting each user to the database server. This allows for increase in performance by allowing ScaleArc access to cache objects without having to connect to the database servers. Turn Windows Authentication Offload ON/OFF.

6.3.2.5 Cache Settings



Sharepoint Application Caching - This setting will exclude the ever changing @RequestGuid correlation parameter's value from the hash generation so that Sharepoint traffic generates more cache hits. This paramenter's value does not contribute to changes in response.

Minimum Data Size for Cache Compression (bytes) - Specify the minimum value up to which the data for a select query response need not be compressed. If the response size is greater

than the value you specify, then the entire response will be compressed when cached. The default and minimum value is 2048 bytes.

Maximum Query Size for Cache (KB) - Specify the maximum value in kilobytes above which the data for a select query response will not be cached. When query cache does not seem to work increase this number to accommodate for larger query responses.

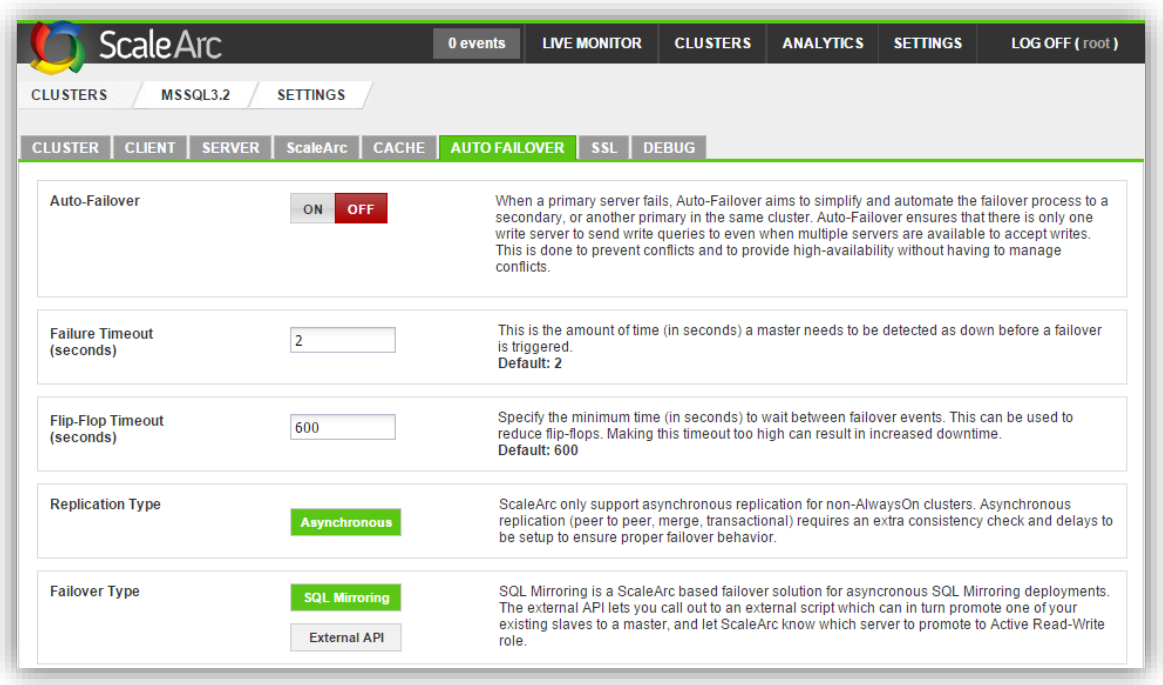
When large queries response stored within ScaleArc, cache memory utilization will also increase (default value is 1024 KB).

Cache Cleanup Cycle Time (seconds) - Specify how often to run the cache garbage collector. This is not related to the minimum time to live (TTL) of your cache. So don't set this to an unreasonably low value. The default value is 60 seconds.

6.3.2.6 Auto Failover Settings

Another advantage of ScaleArc is its Auto Failover feature which aims to simplify the failover process by automating it to a slave, or another master within the same ScaleArc cluster. Auto-Failover ensures that there is at least one write server to send write queries to when multiple servers are available to accept write queries. This is done to provide high-availability, without having to manage conflicts.

Whenever in a particular cluster the last read/write server goes down, failover will upgrade the standby to become the master server. A slave could be added as Standby Traffic and Standby No Traffic. A Standby Traffic is a server that will be used in load balancing under normal operation, but will be promoted to a write server when the last server goes down. A Standby No-Traffic server is meant to receive no traffic during normal operation, until it is promoted to a write server.



Auto- Failover - When a primary server fails, Auto-Failover aims to simplify and automate the failover process to a secondary, or another primary in the same cluster. Auto-Failover ensures that there is only one write server to send write queries to even when multiple servers are available to accept writes. This is done to prevent conflicts and to provide high-availability without having to manage conflicts.

Failure Timeout - This is the amount of time (in seconds) a master needs to be detected as down before a fail-over is triggered. Once the specified time has elapsed with the master as

down, the failover process will be triggered and it will upgrade either a slave or another master to take up the job of the down master server. The default value for this timeout is 0 seconds.

Flip Flop Timeout - This is the minimum time (in seconds) to wait between fail-over events. This can be used to reduce flip-flops and avoid frequent failovers. Although making this timeout too high can result in an increased downtime for the cluster in which case the failover will take place after the specified flipflop timeout. This timeout can be overridden by using the manual failover functionality by which failover is triggered irrespective of the flipflop timeout.

Replication Type – ScaleArc only supports Asynchronous replication for non-AlwaysOn clusters. Asynchronous replication (peer to peer, merge, transactional) requires an extra consistency check and delays to be setup to ensure proper failover behavior.

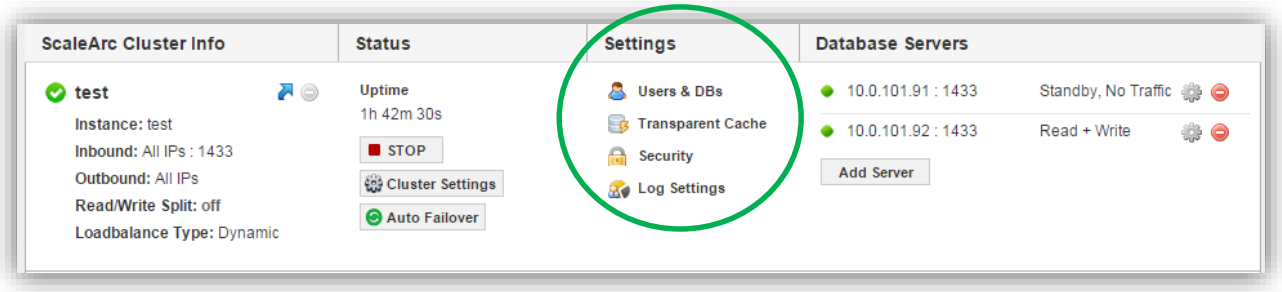
Failover Type – Choose in between two failover types, SQL Mirroring or Extrernal API.

SQL Mirroring – SQL Mirroring is a ScaleArc based failover solution for Asynchronous SQL mirroring deployments. Currently this is the only replication type supported by ScaleArc. For this feature to work databases must be in SQL mirroring replication type.

Pre-requisites:

Following are the pre-requisites for configuring SQL Mirroring in ScaleArc:

1. One of the ScaleArc server should be configured as Read + Write and the other server should be configured as Standby, No traffic.
2. The databases that will be failed over need to be configured in the Users & DB's section within a cluster.
3. At least one of the databases between primary and secondary server must be configured as Mirroring and same should be configured within Users & DB's section
4. First database configured in Users & DB's section within ScaleArc must be configured as Mirroring.



Benefits:

Following are the benefits of SQL Mirroring in ScaleArc:

- Maximize resource utilization and save cost through
  - Enabling automatic failure without an additional witness server.
  - Avoiding the cost of witness server setup and maintenance.
- Improve application performance/uptime by
  - Enabling automatic failure without an additional witness server
  - Routing traffic around downed servers - even across multiple data centers for maximum availability.
  - Enabling application transparent automatic server failover
  - Handling auto failover in asynchronous operation mode.
  - Zero-downtime system migration and upgrade
- Avoid application errors
  - Routing traffic around downed servers - even across multiple data centers for maximum availability.
  - Enabling application transparent automatic server failover
  - Handling auto failover in asynchronous operation mode.

  - Zero-downtime system migration and upgrade
- Improve administrative efficiency and avoid manual setup/deployment by:
  - Providing automatic failure without an additional witness server
  - Handling auto failover in asynchronous operation mode.
  - Zero-downtime system migration and upgrade
  - Effortlessly manage servers hosting the databases through ScaleArc

Limitations of SQL Mirroring in ScaleArc

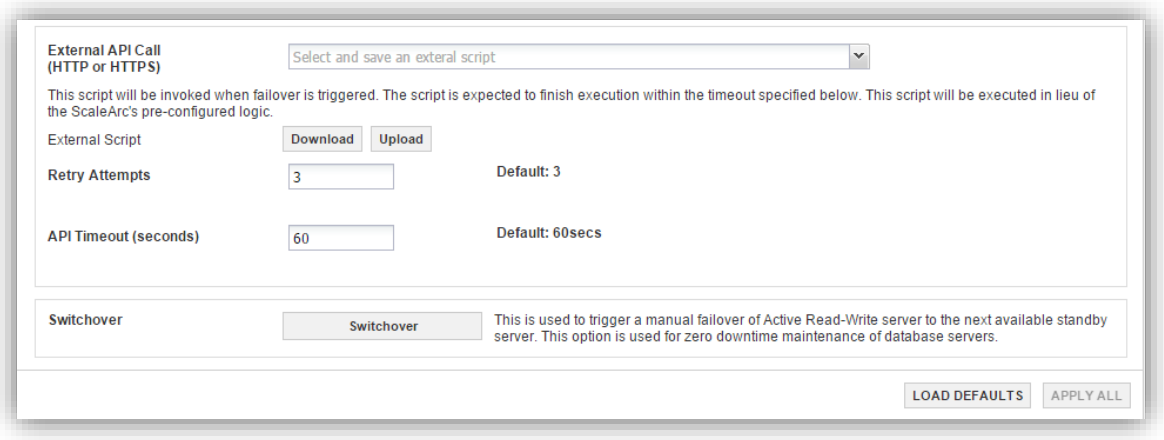
The following section describes the limitations of SQL Mirroring in ScaleArc:

- As a note of caution, if any database configured in Users & DB's are in suspended mode then ScaleArc aborts the switchover process.
- In a situation where the primary server is healthy but ScaleArc is not able to reach the primary server, then ScaleArc will not perform failover.

External API based - The external API lets you call out to an external script which can in turn promotes one of your existing slaves to a master, and lets ScaleArc know which server to promote to Active Read-Write Role.

External API Call – Select the external API script from the drop down list.

External Script – Upload/ Download an external API script.



The script will be invoked when failover is triggered in place of ScaleArc's pre-configured auto failover logic. The script is expected to finish execution within the **API timeout** specified.

Retry Attempts – Select the number of retry attempts.

Switchover - This option can be used to trigger a manual failover of active read-write server to the stand by server. Also this functionality overrides all the others specified by the ScaleArc failover and does a manual failover as per customer requirements. Specify the maual failover timeout. The active connections in intermediate state that do not complete in the specified time will not be switched to a Standby server. Active connections which are not in intermediate state will be switched to a Standby server immediately.

6.3.2.7 SSL

The SSL Settings for a cluster can be modified from this section.

You can enable/ disable SSL settings for a particluar cluster.

Note: Any modification to the SSL Offload option would require a cluster restart.

This screen shows you the various certificates uploaded during cluster creation.

Use this screen to edit/ delete client and server certificates.

SSL offload allows you to establish secure communication between the client and the server via ScaleArc.

Important - as a prerequisite, if a SQL client needs to initiate full encryption with ScaleArc's SSL enabled cluster, ScaleArc inbound IP should be reverse resolvable to the hostname entry in certificate uploaded on ScaleArc cluster.

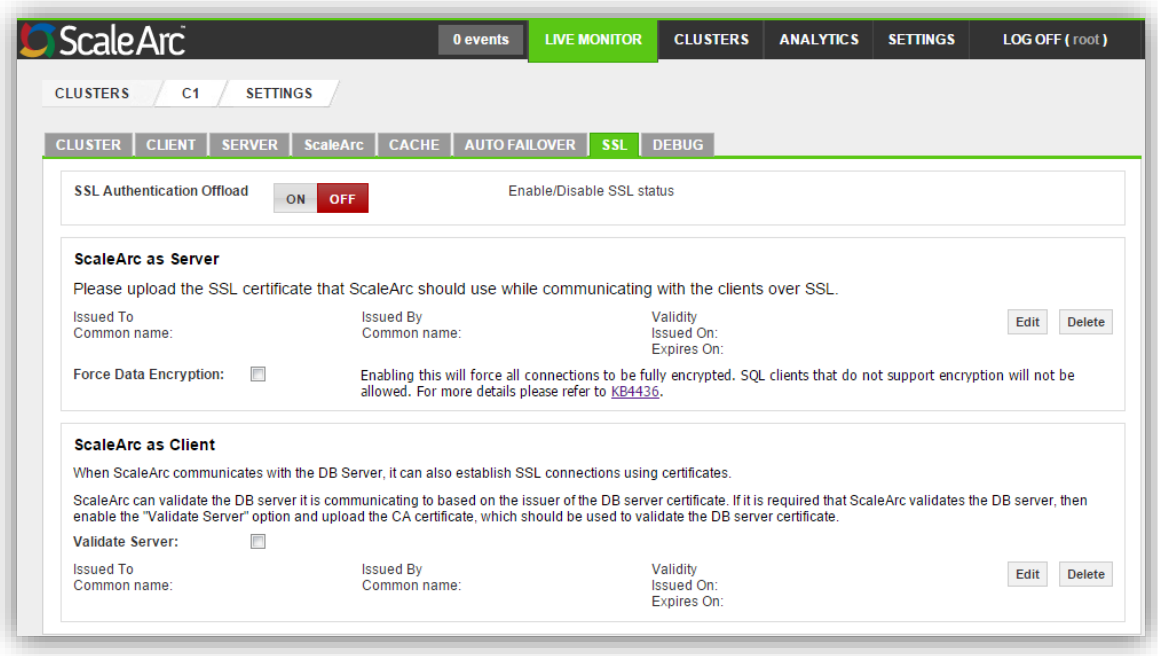
If you wish to enable SSL Offload then ScaleArc would require a certificate and a key pair to be uploaded. Upload the **`Server certificate`** by clicking on **Browse**. Upload the **`Server key`** as well in the same manner.

Enabling **`Force Encryption`** will force all connections to be fully encrypted. SQL clients that do not support encryption will not be allowed.

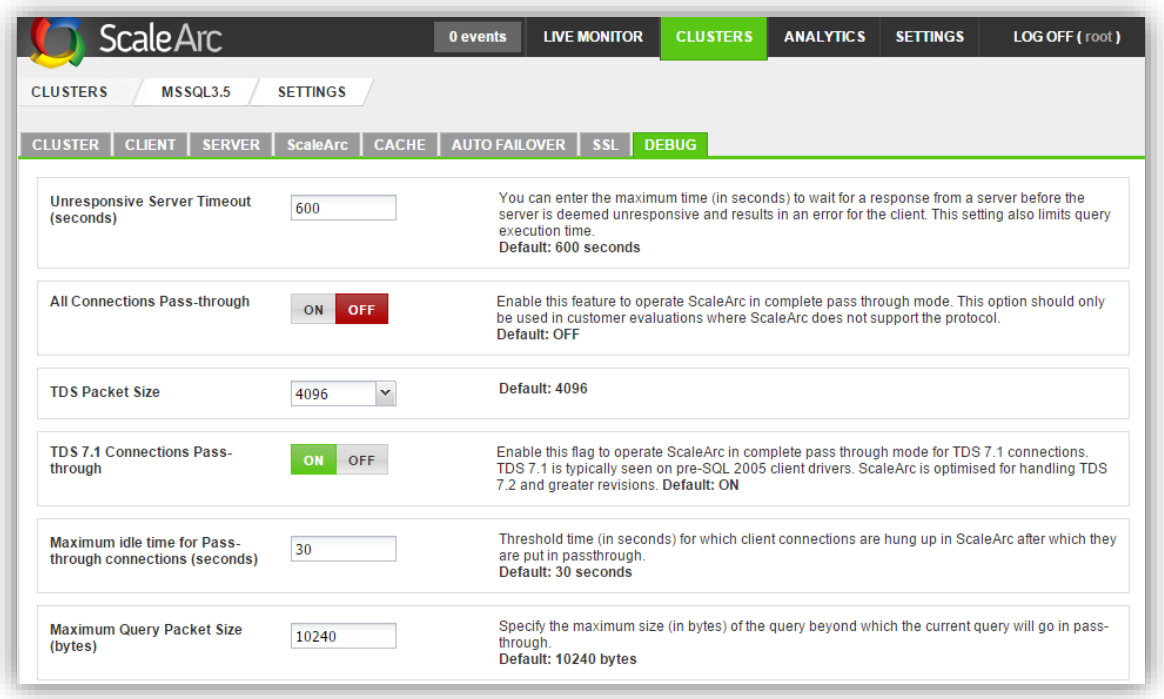
`Validate Client` allows ScaleArc to validate the clients connecting to ScaleArc based on their issuer. If you require client validatation then enable this option and upload the appropriate **`CA certificate`**.

ScaleArc can also connect to the SQL Server using client certificates. If you require secure communication between ScaleArc and the DB server then enable the **`Client certifcate`** option and upload the client certificate and key pair.

Similarily select the **`Validate Server`** option if you require ScaleArc to validate the DB server communication and upload the appropriate CA certificate.



6.3.2.8 Debug Settings



Unresponsive Server Timeout (seconds) - This setting is inactive if the fail-safe status is ON. When fail-safe queries are OFF you can enter the maximum time to wait for a response from a server before the server is deemed unresponsive and results in an error for the client. (Default value is 600 seconds)

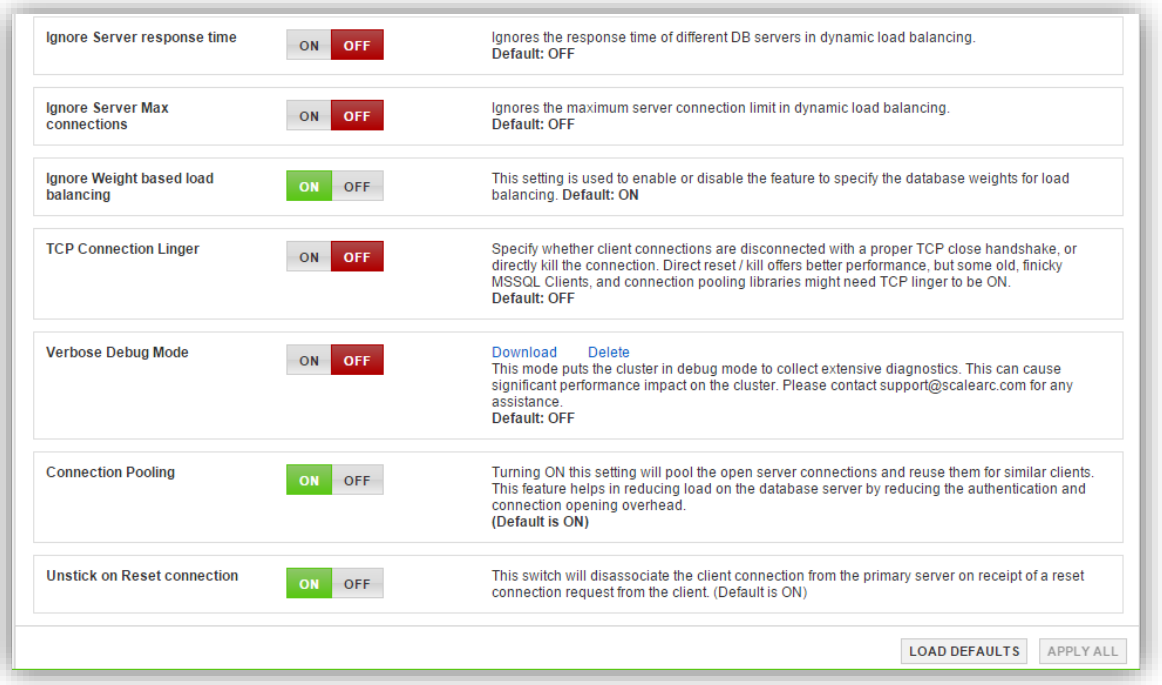
All Connections Pass-through - Enable this feature to operate ScaleArc in complete pass through mode. This option should only be used in customer evaluations where ScaleArc does not support the protocol.

TDS Packet Size – You can configure the SQL Server packet size, which is the size of the TDS packets. Default is 4096 bytes. A cluster restart is required if you make any changes to the TDS packet size.

TDS 7.1 Connections Pass-through - Enable this flag to operate ScaleArc in complete pass through mode for TDS 7.1 connections. TDS 7.1 is typically seen on pre-SQL 2005 client drivers. ScaleArc is optimized for handling TDS 7.2 and greater versions.

Maximum Idle Time for Pass-through Connections - The time (in seconds) after which if ScaleArc does not receive any response from the server, the connection is marked as pass-through.

Maximum Query Packet Size (bytes) - Specify the maximum size of the query beyond which the current query will go in pass-through.



Ignore Server response time – When this setting is enabled, ScaleArc will ignore the response time of DB servers in dynamic load balancing.

Ignore Server Max Connections – When this setting is enabled, ScaleArc will ignore the maximum server connection limit ver settings for dynamic load balancing.

Ignore Weight based load balancing - This setting is used to enable or disable the feature to specify the database weights for load balancing.

TCP Connection Linger (ON/OFF) - Specify whether client connections are disconnected with a proper TCP Close Handshake, or directly kill the connection. Direct reset / kill offers better performance, but some old, finicky MS SQL Clients, and connection pooling libraries might need TCP Linger to be ON.

Verbose Debug Mode (ON/OFF) – This setting is used to enable/ disable the troubleshooting mode in ScaleArc on a cluster level. This mode will capture and download the raw logs (including tcpdump) for the cluster. Enabling this option will create large log files and may slow down performance. Please contact ScaleArc support if you require assistance.

Click on **Download** to begin downloading the log files. Specify the duration that you wish to execute the verbose debug mode on a cluster. The verbose debug mode will generate an event for the log files. The downloaded log files are available in tar.gz format. Only the latest debug file will be available to download. Click on **Delete** to delete the log files.

Please note, the cluster must be running to enable the verbose debug mode. The maximum file size for tcpdump files is 1GB or a maximum of 10 files of 100MB each.

Connection Pooling (ON/OFF)– Enabling this feature allows you to pool the open server connections and reuse them for similar clients. This helps in reducing the number of connections to the servers by reducing the authentication and connection opening overhead which in turn increases server performance and availability.

Unstick on Reset Connection (ON/OFF) - This switch will disassociate the client connection from the primary server on receipt of a reset connection request from the client. Default is ON in Read-Write split mode.

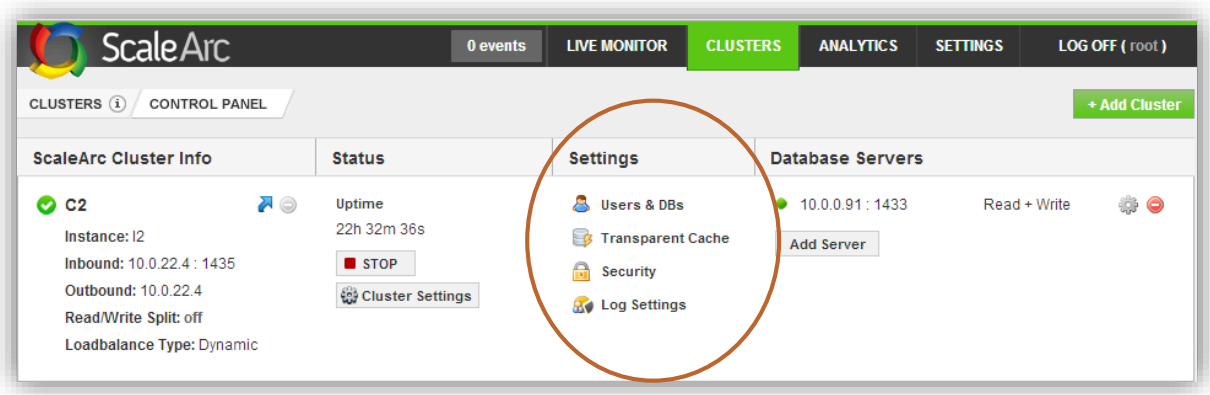
6.3.3 Settings

Cluster settings consist of 4 sections. Let's take a look at each section in detail. Users defined here are authenticated by the DB server for SQL transaction execution.

6.3.3.1 Users & DBs

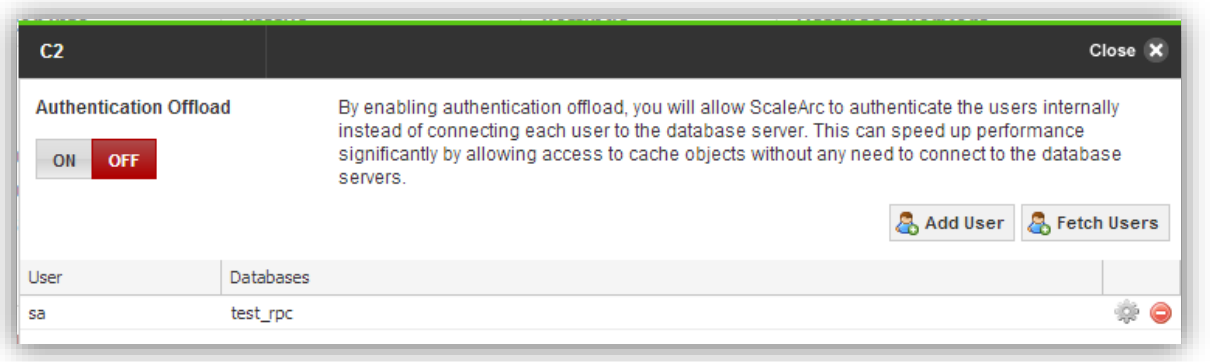
Click on **User & DB's** to see a list of users associated with a cluster.

Refer to the **user management** section of this manual for creating a user.



Authentication Offload

In this section we can enable or disable authentication offload. Enabling authentication offload can speed up performance significantly. The bi-product of this feature provides connection pooling and multiplexing. Client sessions terminate on ScaleArc instead of the database. ScaleArc provides session management by reducing the total number of sessions to the DB servers which in turn increase performance.

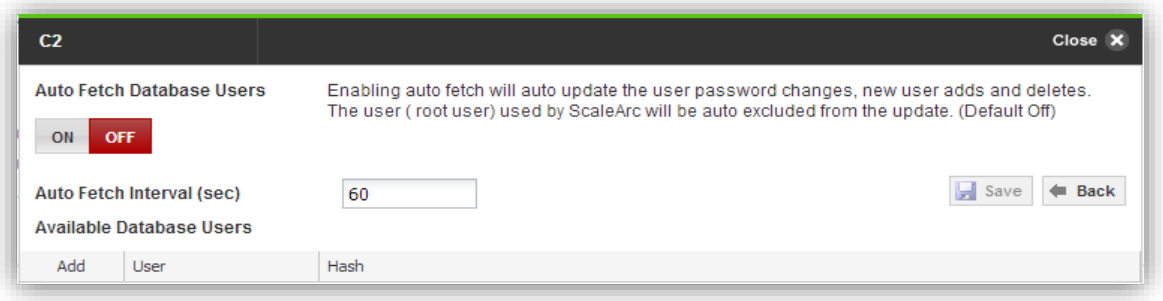


Configure – Clicking on the gear icon allows you to configure access to / control cache for a particular user. Any databases not specifically added here may not be fully accessible.

Delete User – Click this option to delete a user added on ScaleArc for the cluster.

Add User – Clicking on **Add User** allows you to create a new user and specify the databases you would like to provide access to or control cache. Click on **Save** to save the user.

Fetch Users – Click on Fetch Users to enable auto fetch which will automatically add the databases that the users have permissions to.



Auto fetch will also auto update any modicfications to users. You can set the interval when auto fetch will run.

6.3.3.2 Transparent Cache

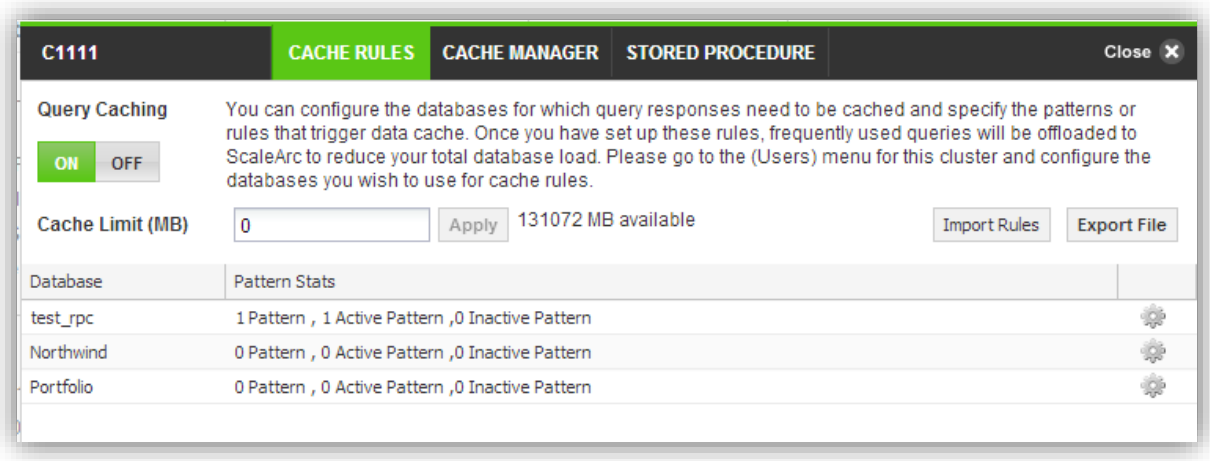
Cache Rules

Cache rules are defined per cluster. These are simple patterns or expressions for ScaleArc to parse against and if matched serve directly from ScaleArc's memory. ScaleArc creates a hash map similar to that of memcached for which can achieve up to 12x faster performance than conventional database servers.

Query Caching – Query Caching can be explicitly turned OFF or kept ON (ON by default).

Cache Limit (in MB) - Maximum cache limit can be set from here. 0 is the default which means unlimited up to the license limit.

A list of all databases is listed with summary statistics illustrating total number of patterns and how many of those patterns are active and inactive.



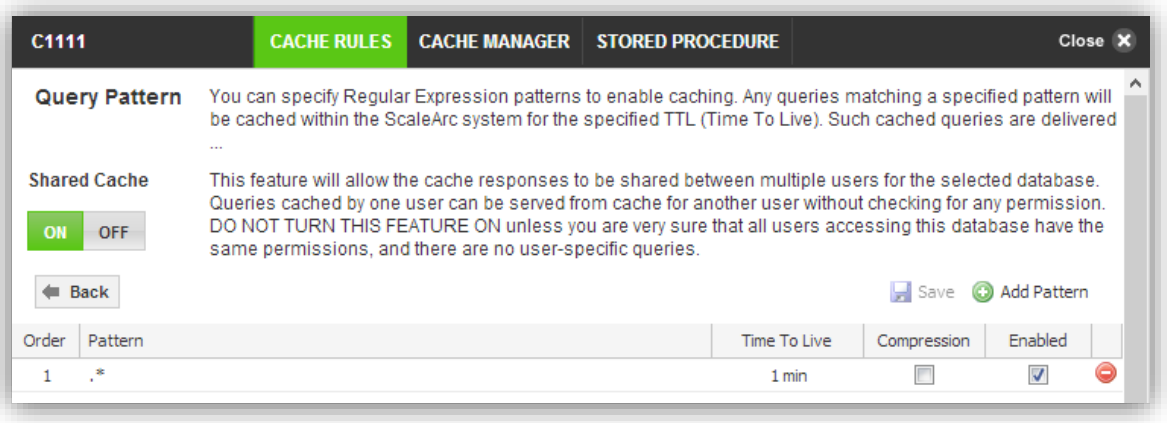
Entering cache rules to ScaleArc is made simple and flexible. This step demonstrates a manual entry. Cache patterns can also be created with a one-click option through **analytics** from real patterns of traffic that have passed through ScaleArc. The API can create cache rules as well and parse in direct through code with a prefix to the SQL statement.

Export File - This feature enables exporting all the cache rules into a csv file so that it could be imported to another ScaleArc or cluster easily.

Import Rules – This feature enables importing the cache rules from a csv file that was generated from ScaleArc. This saves the effort to recreate the cache rules easily if they are already created on another ScaleArc cluster or instance.

Click on the **Edit** (gear) button next to the query patterns. From this screen you can view and define query patterns with regular expressions that filter the traffic, which passes through the ScaleArc interface. Any queries matching a specified pattern will be cached within the ScaleArc appliance for the specified TTL (time to live). Such cached queries are delivered to the application directly by the ScaleArc appliance significantly faster than the database and improves the overall performance of your application.

You can read more about how regular expression patterns work here.



Pattern - Patterns can include any Perl Compatible Regular Expression. We typically see wild card expression to create simple rules. For example,.*6.* adds into cache any string with a value of 6 in the string.

Time to Live – Defines when the cache expires.

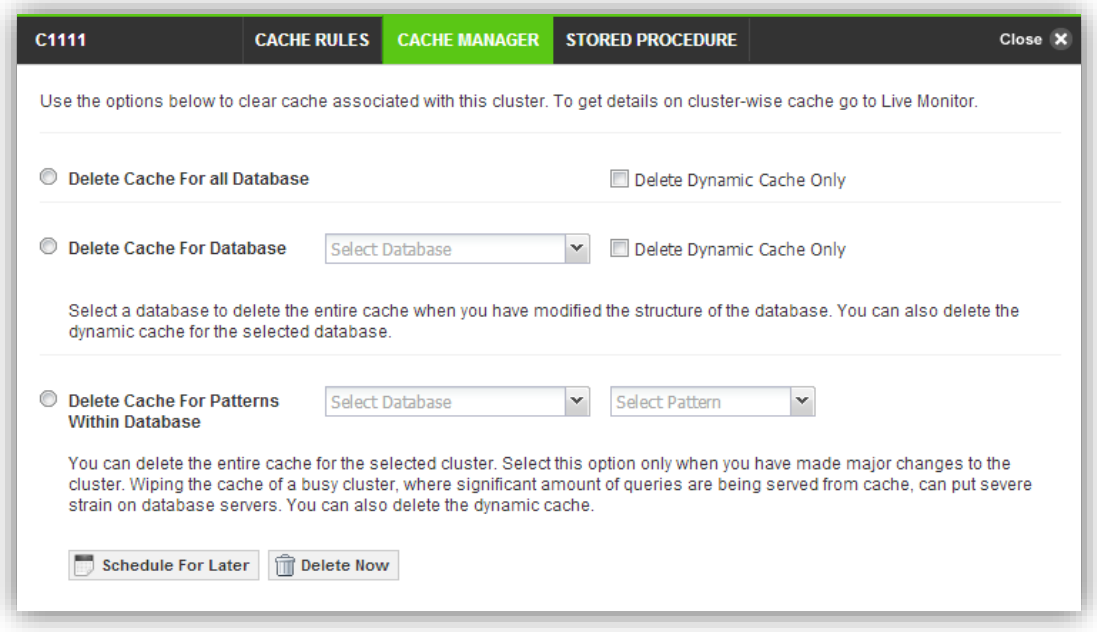
Compression (ON/OFF) – This enables compression for any cache entry. Maximum and minimum compression levels can also be modified in the clusters **advanced settings**. The default setting is OFF.

Enabled (ON/OFF) - Disable or enable a cache rule from executing depending on its state. The default setting is ON.

Edit/Delete – Clicking on the gear icon will allow you to make changes to the cache rule. You can also click on the delete icon to delete the cache rule.

Cache Manager

Cache manager allows you to clear the cache from memory under very specific circumstances. Standard cache control expire with a TTL setting however there are conditions when we need more control over dynamic content or content that may expire periodically.



Here we can delete all cache, delete only dynamic cache, specify if cache is specific to a database which matches a specific pattern and apply these rules to a schedule.

Delete Cache For all Database – This option deletes the entire cache for the selected cluster. Select this option only when you have made major changes to the cluster. Wiping the cache of a busy cluster, where significant amount of queries are being served from cache, can put severe strain on database servers.

Delete Cache For Database – Select a database to delete the entire cache when you have modified the structure of the database or added or removed major tables. You can also delete the dynamic cache for the selected database.

Delete Cache For Patterns within Database – Select a database and the pattern to delete the entire cache for the pattern added for the specified database. A single pattern may be responsible for caching many queries.

Warning - Wiping the cache for a pattern will result in additional load on the database server for a compute-intensive query which matches that pattern.

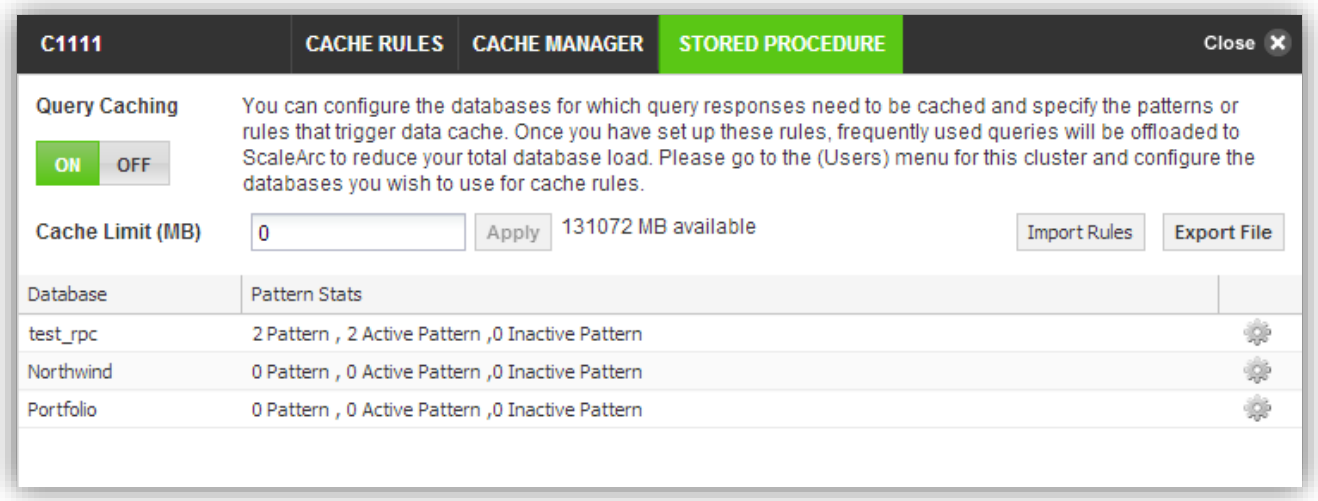
Click on **Schedule** to schedule the deletion of the cache at the specified hour of the day.

Click on **Delete Now** to delete the cache.

Click on **Cancel** to exit out of the screen without making any changes.

Stored Procedure

From this screen you can configure the databases you wish to set up rules to trigger data cache. Query Caching can be turned OFF or kept ON (ON by default).



Cache Limit (in MB) - Maximum cache limit can be set from here. 0 is the default which means unlimited up to the license limit.

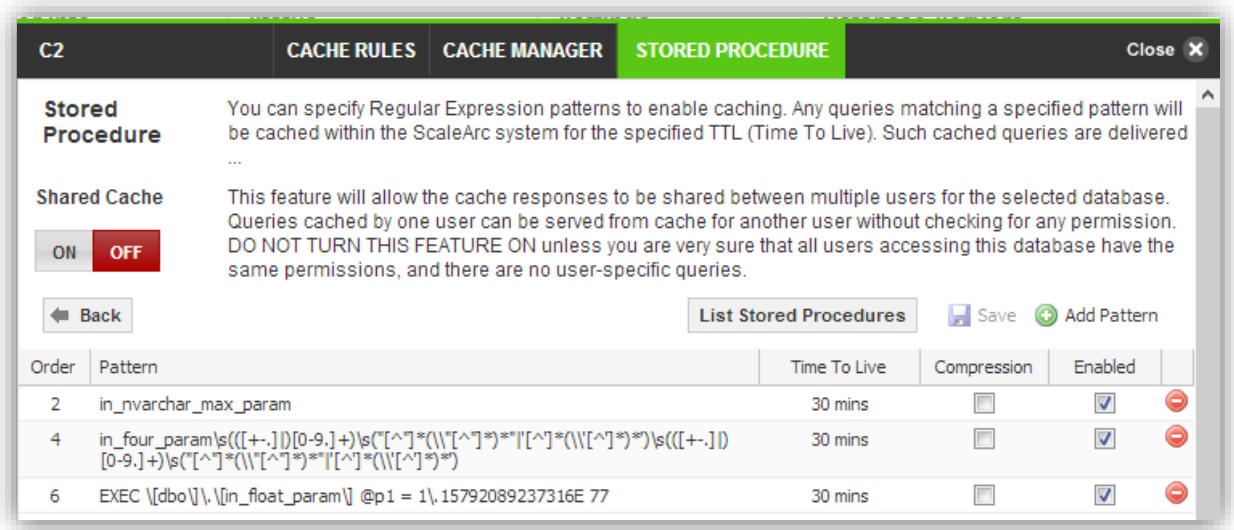
A list of all databases is listed with summary statistics illustrating total number of patterns and how many of those patterns are active and inactive.

Entering cache rules to ScaleArc is made simple and flexible. This step demonstrates a manual entry. Cache patterns can also be created with a one-click option through **analytics** from real patterns of traffic that have passed through ScaleArc. The API can create cache rules as well and parse in direct through code with a prefix to the SQL statement.

Export File- This feature enables exporting all the cache rules into a csv file so that it could be imported to another ScaleArc instance or cluster easily.

Import Rules – This feature enables importing the cache rules from a csv file that was generated from ScaleArc. This saves the effort to recreate the cache rules easily if they are already created on another cluster or ScaleArc instance.

Click on the **Edit** (gear) icon to view the caching patterns. Any queries matching a specified pattern will be cached within the ScaleArc system for the specified TTL (Time To Live).

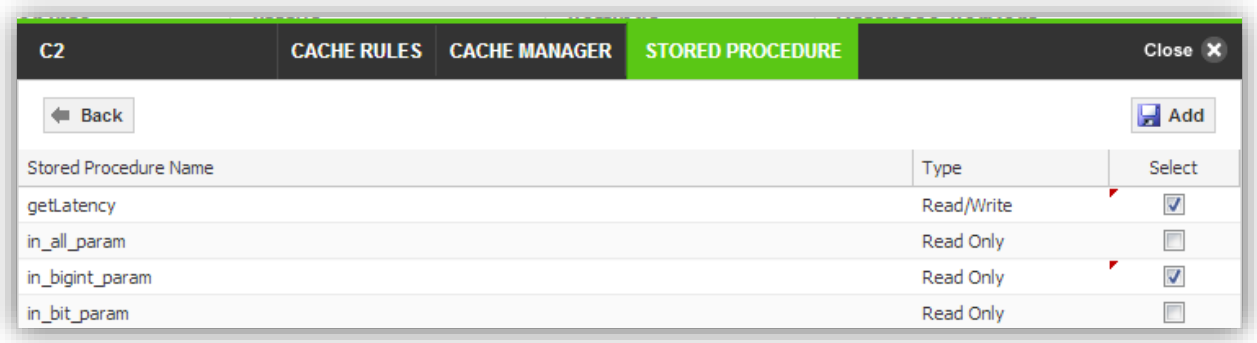


Shared Cache – Turn Shared Cache feature ON to allow cache responsed to be shared between multiple users for a particluar database.

Warning – Turn this features ON only if you are sure that all users accessing the database have the same permissions.

Add Pattern – Add patterns to be cached.

List Stored Procedures - List of patterns stored on the server. Select the ones you would like to add as a part of shared cache patterns.

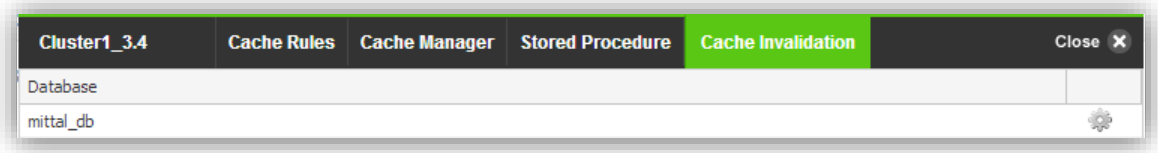


Cache Invalidation

This feature automatically invalidates a cache entry based on the updates occuring on the database. You can add invalidation patterns to the system and map these invalidation patterns to caching patterns by making them belong to the same group. When the invalidation pattern matches, it would invalidate the cache patterns which belong in the group. Create a logical group of invalidation patterns and caching patterns from the screen below.

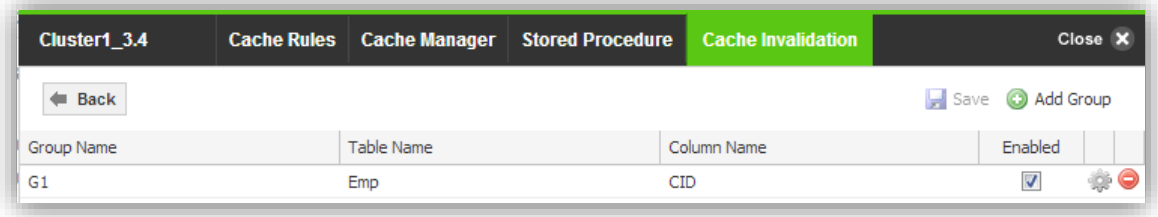
Click on the **Edit** (gear) icon to add or configure caching patterns for the database.

Note: A database for the user must be configured in order to add caching patterns.



You can add additional groups by clicking on the

option. You can add/edit the patterns within the existing groups by clicking on the gear icon.

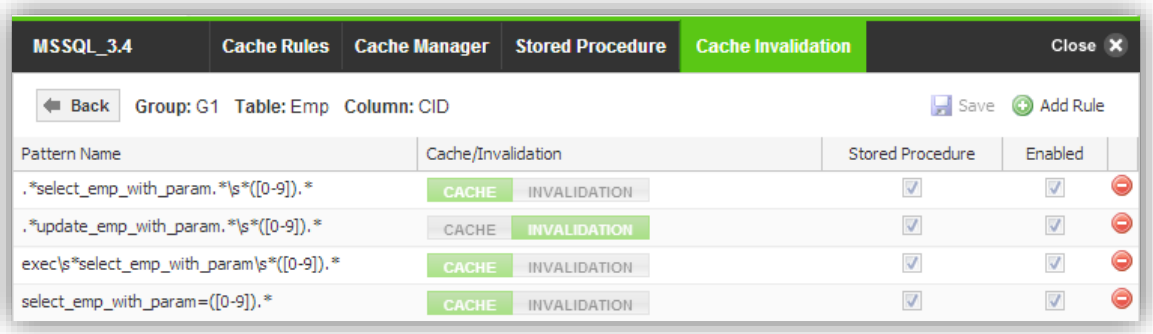


The screen below shows the rules within the group. It also highlights whether the pattern is a caching or an invalidation pattern.

Click on **Add Rule** to add patterns for caching and invalidation which will belong to this group. Specify the pattern and select the pattern type. Also you may choose to enable/ disable the rule. Also you may choose to add stored procedure caching pattern as well.

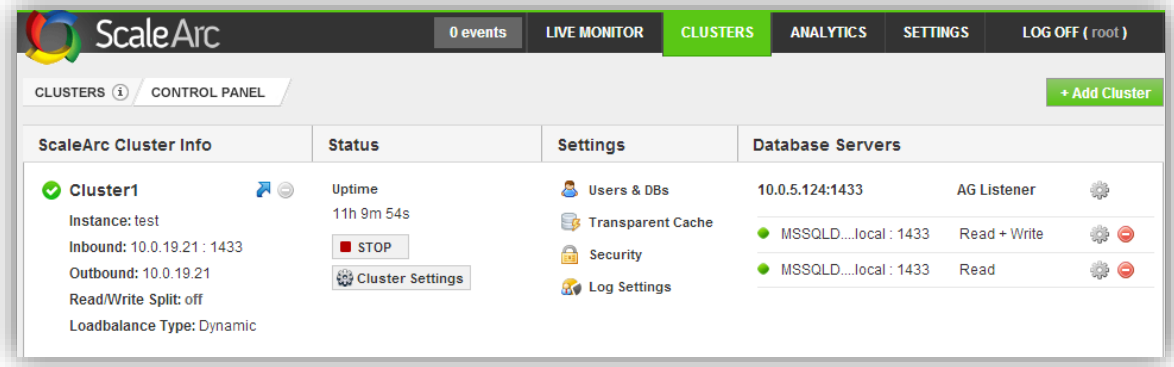
While adding a query pattern to a cache invalidation group remember the table name and the column name must be present in the pattern and the pattern regex should fetch the column value. This is ensured by having '()' around the column value. The regex pattern should be constructed keeping these rules in mind.

Similar rules apply for an invalidation pattern. For stored procedure pattern it is important that the parameter that depicts the column value should be within '()'.
Ensure that a cache pattern which belongs to a group should be unique across all patterns present either in normal cache rules or other groups.



6.3.3.3 Security

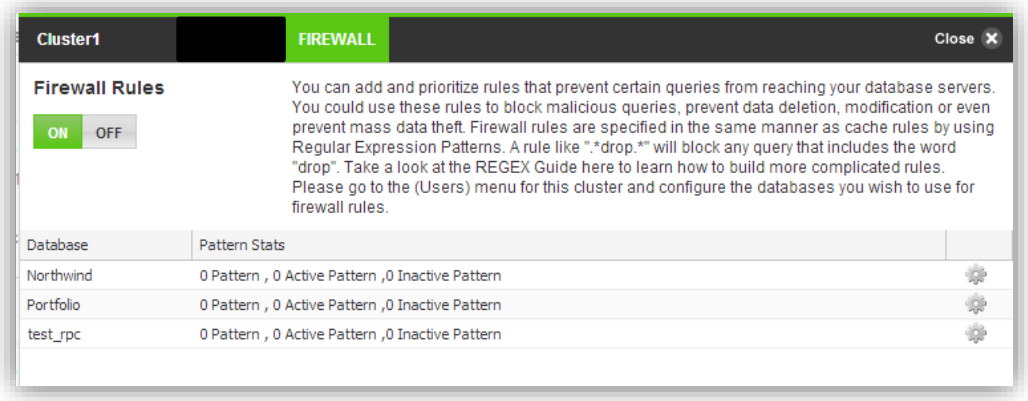
The third option under cluster settings is Security. From here you can define Firewall rules for the cluster.



Firewall - Query firewall rules work very similar to cache rules in the way it filters on SQL string using patterns and expressions.

You can add and prioritize rules that prevent certain queries from reaching your database servers. You could use these rules to block malicious queries, prevent data deletion or modification, or even prevent mass data theft. Firewall rules are specified in the same manner as cache rules by using regular expression patterns. A rule like ".*drop.*" will block any query that includes the word "drop".

Take a look at the **REGEX guide** to learn how to build more complicated rules.



The query firewall rules screen lists all databases and the patterns associated with a database. The databases that contain patterns are highlighted in light grey.

Click on the edit icon to open the database and its list of rules.

You can turn a firewall rule for a cluster **ON** or **OFF.**

You can add a new firewall rule by entering in a rule and clicking on **Add.**

Pattern – Lists out all the firewall rules for the database selected.

Enabled – You can enable or disable the rule.

Delete - Clicking on the delete icon will remove the rule from database.

Click on **Save** to save changes or **Cancel** to exit out of the screen without saving the changes.

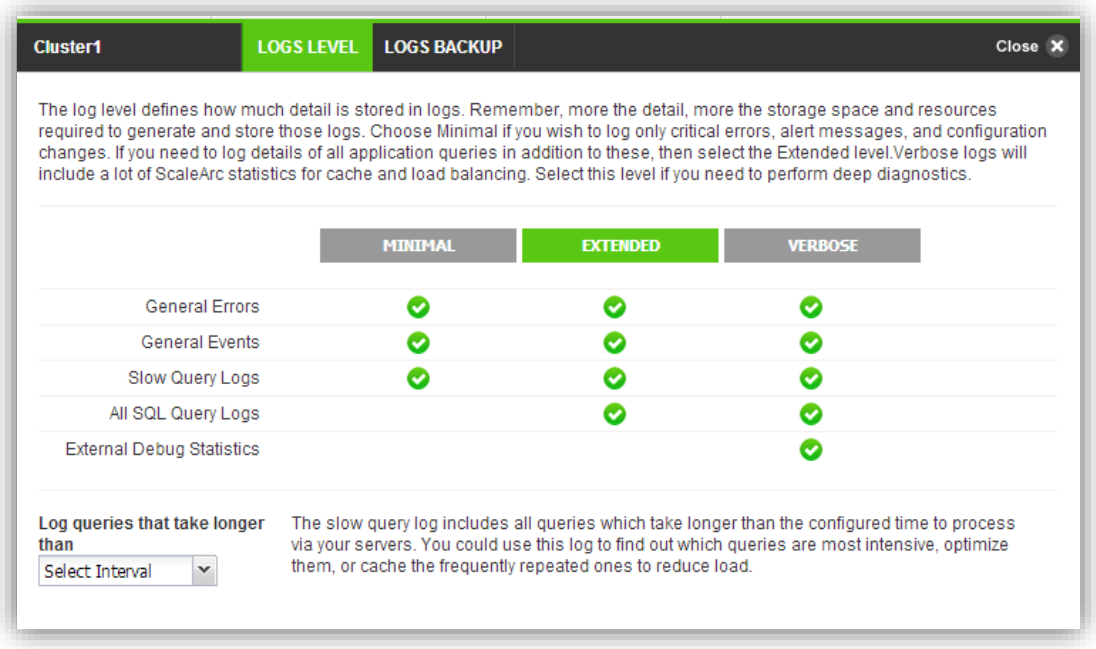
6.3.3.4 Log Settings

Log setting controls the log level ScaleArc maintains for the each cluster.

Logs Level

The logs level defines how much detail is stored in logs.

Note: The more detail you store in logs, more storage space and resources will be required to generate and store those logs.



Minimal logging - Writes critical errors, alert messages and configurations changes.

Extended logging - Includes everything from minimal logging and adds application queries. This is the default setting.

Verbose logging – Includes everything from minimal, extended and adds cache and load balancing statistics. See **real time logs and log tailing** for samples.

Log Queries that take longer than - Captures queries that exceed a configured performance threshold and writes them to a specific slow query log file.

Logs Backup

Uploads all log files to an ftp server to maintain historical log files beyond the storage capacity of the appliance.

Backup logs to FTP every – Select the interval you want your logs to be dumped to a FTP server.

FTP Host – IP address of the FTP server.

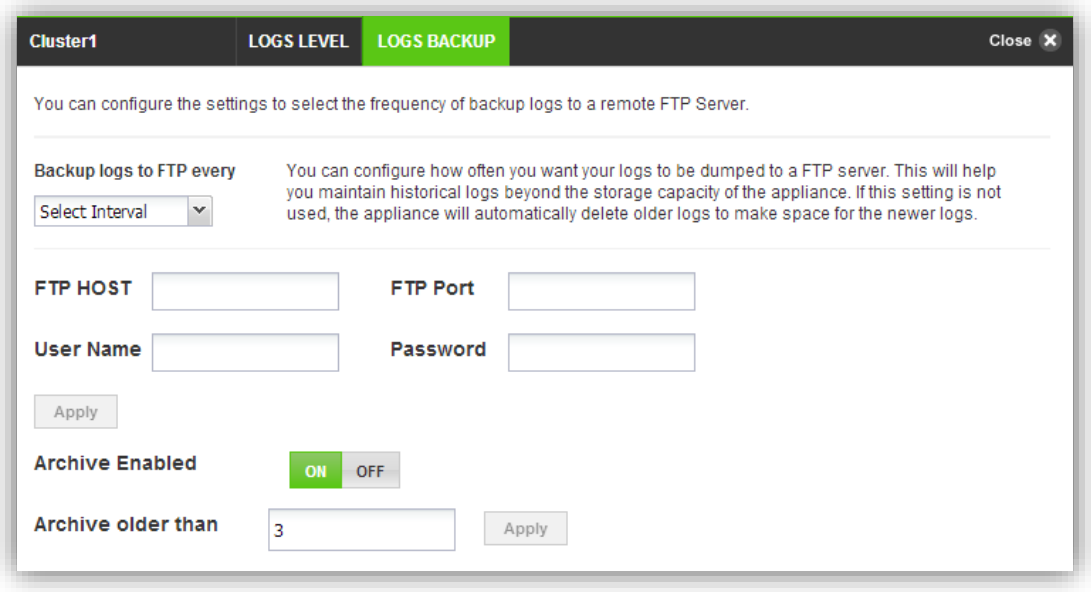
FTP Port – Port of the FTP server.

Username – Username to login into the FTP server.

Password – Password of the username specified above.

Archive Enabled – Select whether or not you wish to archive log files.

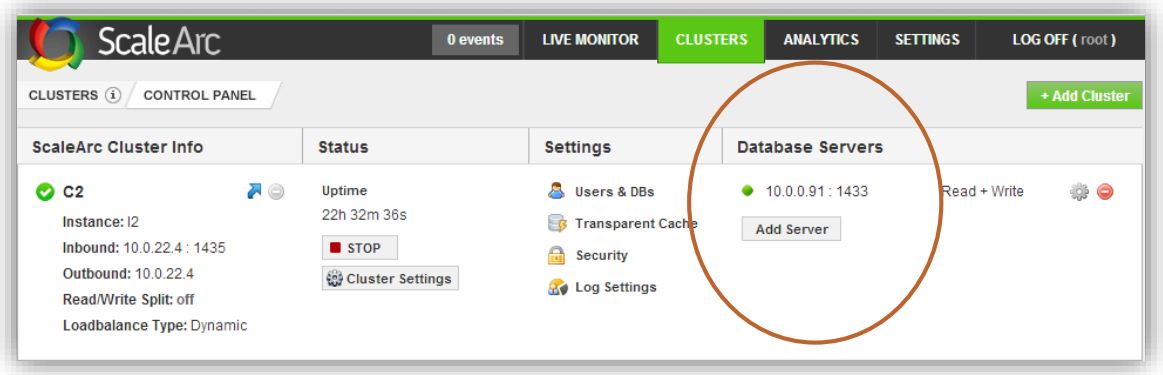
Archive older than – Enter the number of days log files will be archived after in this area.



6.3.4 Servers

In ScaleArc, servers are your database servers. For any one cluster, we must have at least one server associated. ScaleArc uses the servers listed in its load balance rotation. Additionally, ScaleArc validates the integrity of each server and the consistency of the database as it relates to replication.

The number of servers ScaleArc supports is determined by the license file applied to ScaleArc. ScaleArc benchmarks performance for listing adequate resources to match traffic, application behavior and database performance.



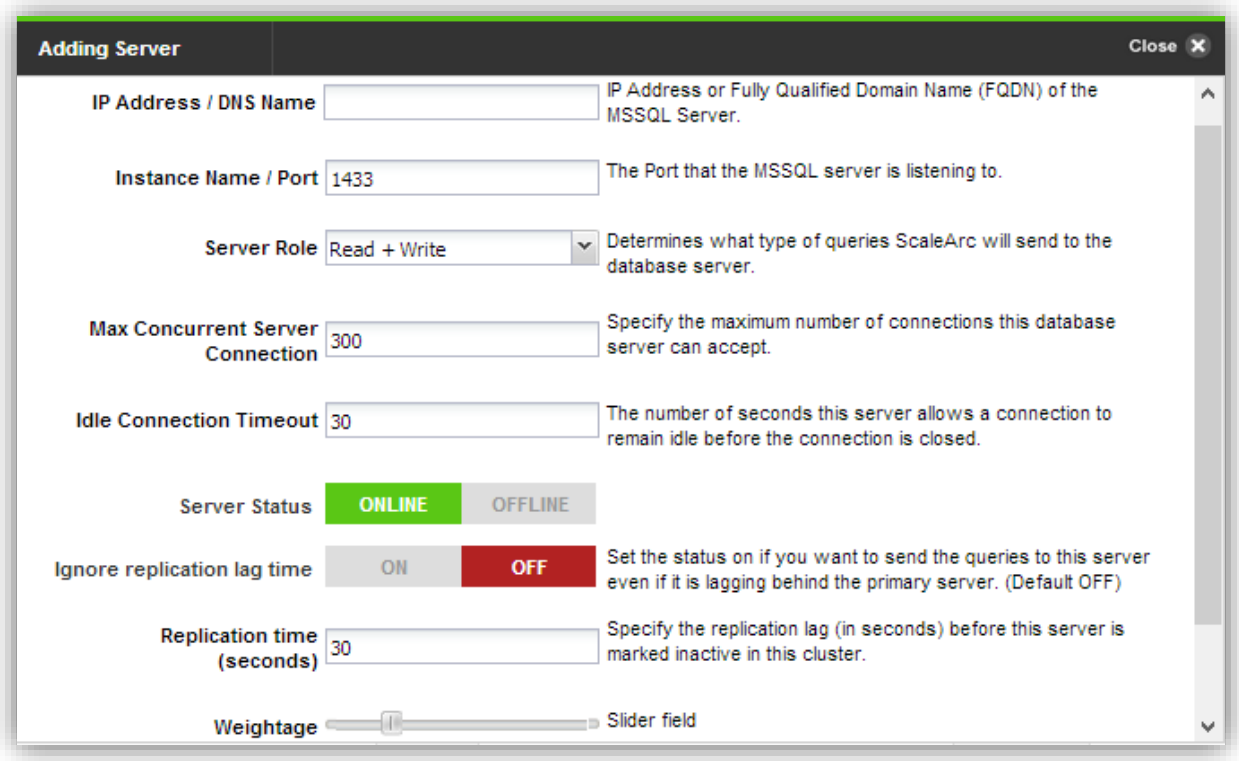
In the figure above we see the list of servers configured for the cluster. These servers where added when we first created the cluster through the **create a cluster** wizard.

Once a cluster is created, we can modify the underlying servers supporting the cluster. This includes:

- Add servers into the cluster
- Delete a server from a cluster
- Modify a server's properties in a cluster

Let's add a server into the cluster and explore the options in more detail. All options are available for modification once the server has been added to a cluster.

When we select the **Add Server** link a form page opens with the following fields:



IP Address/ DNS Name - IP address of the database server**.**

Instance Name/ Port - Database servers listening TCP port.

Sever Role – Lets ScaleArc know what servers support reads or read and writes. This parameter is used when transparently splitting read/ write queries between the various clusters on the server.

- ***Read + Write*** (typically a master database server)
- ***Read*** (typically a slave/ replica database server)

Maximum Concurrent Server Connection – Lets ScaleArc know what the maximum number of connections a database server can accept. This supports SQL surge queue and affects the load balance rotation scheme ScaleArc used to distribute traffic across multiple servers.

Server Status – Specificy whether the server is online/offline.

Ignore replication lag time – Set the status ON is you want to send queries to the server even if it is lagging behind the primary server.

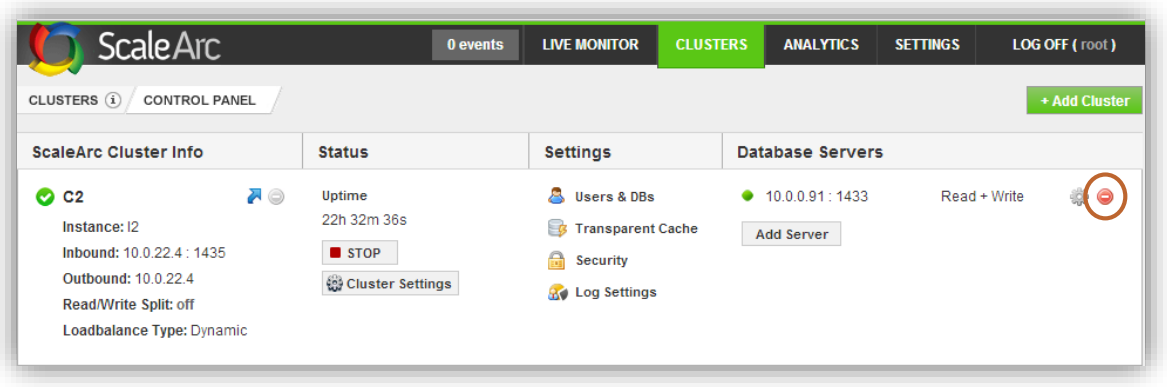
Replication time – Specify the replication lag (in seconds) before this server is marked inactive in this cluster.

Weightage – Use this slider to adjust the amount of connections that a database server can handle. Higher the weightage, larger the numbers connections may be received by the database server.

Click on **Add Server** to save the changes made or **Cancel** to exit out of the screen.

The nature of ScaleArc's load balancing functionality allows it to be dynamic to availability. Database server status of unavailable, being out of replication or time sync with the master will cause ScaleArc to remove the database from the load balance rotation. Database server status is indicated by the green or red status icon next to its IP address. If the condition is red you should be alerted that this Database server is not included in the load balance rotation for this cluster and should be investigated.

Deleting a server from a cluster is as simple as selecting the delete icon next to the server instance.



A few things to remember:

Read/Write Split requires two servers or more to be enabled. ScaleArc will automatically adjust and offer options based on whether there is a single server or multiple servers in the cluster configuration. We can turn off read/write split in the cluster properties. See **edit a cluster** section of this guide **for more information.**

Max Concurrent Client Connections should not exceed the configured value supported on the database server. ScaleArc will provide a warning if the setting in ScaleArc exceeds the threshold configured on the database server. We recommend configuring ScaleArc slightly below the maximum connection count by 80%-90%.

Reducing the Max Concurrent Client Connections to 0 is not recommended. If you need to remove a server from the ScaleArc load balance rotation, delete it or just take the database offline. If you take the database server offline ScaleArc will adjust dynamically.

Read/ write split settings in ScaleArc and its impact on server roles within ScaleArc clusters: For Typical – Master Slave environments read/write split has to be turned ON so that ScaleArc can send reads to slaves and everything else (+ some read traffic) to masters.

Configuring a MS SQL server as a slave does not prevent the slave server from receiving write traffic and acting on it. ScaleArc will only warn against such configurations only at configuration time and not at runtime when traffic is passing.

Here is the matrix with the read/write split and its impact on server roles/types with the corresponding ScaleArc behavior :

Server types in the cluster	Read - Write split ON	Read - Write split OFF
All servers in the cluster are Read-Only	Writes will go to the surgeQ and reads will go to slaves. This mode should be used when switching between masters and there is a short window of time where all servers will be read-only. **Not advised for any production traffic.**	In this mode all servers in the cluster are treated equal. Writes coming in can be sent to any server in the cluster. Use only when the application traffic is truly complete read-only traffic.

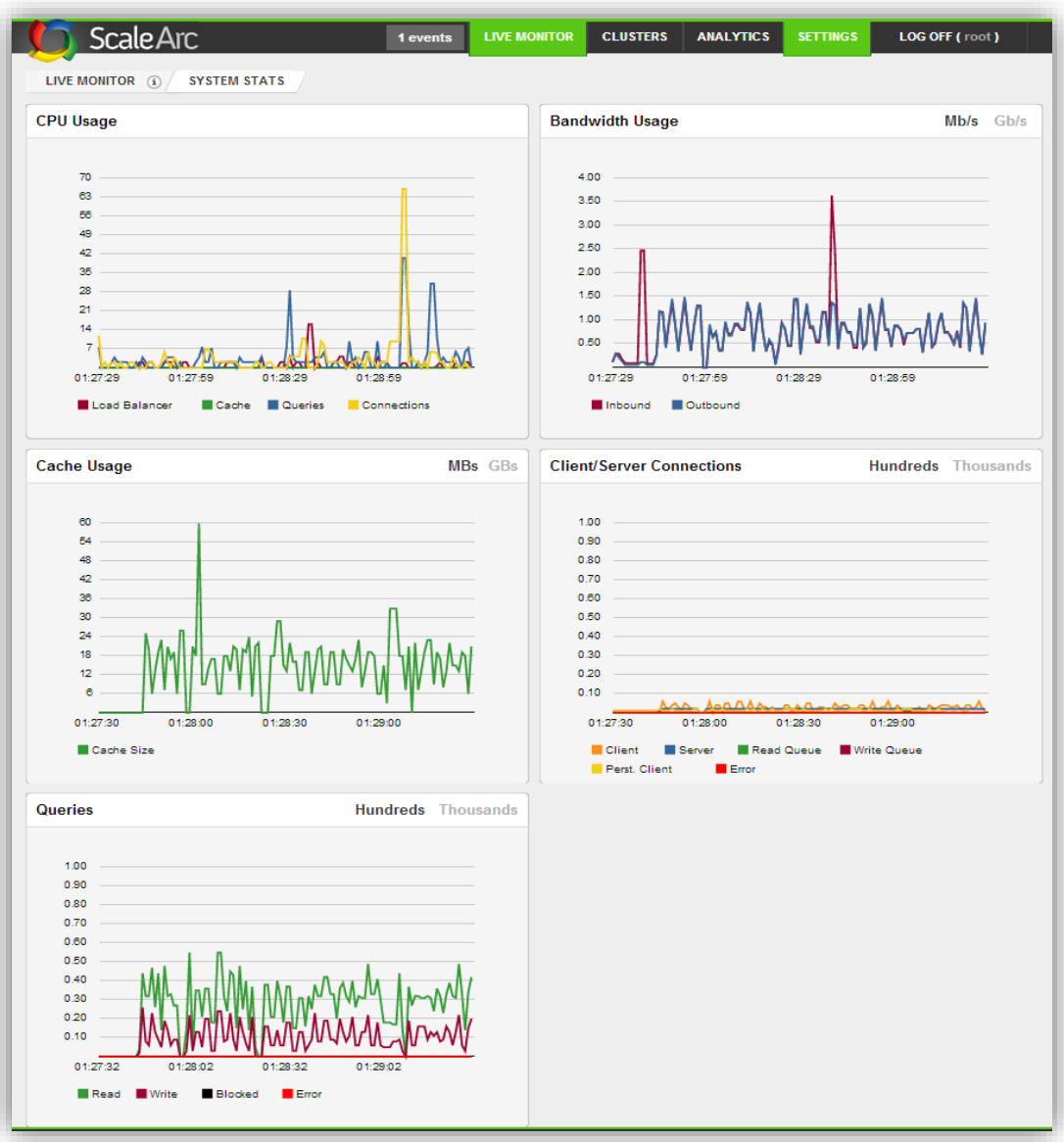
One Read-Write and multiple Read-Only slaves	Typical deployment for production traffic. ScaleArc sends reads to slaves and everything else (+ some read traffic) to master (read-write server).	In this mode all servers in the cluster are treated equal. Writes coming in can be sent to any server in the cluster including read-only servers. **Not advised for any production traffic.**
Multiple Read-Write servers. No Read-Only servers	No impact of read/write split. All traffic will be sent to all servers.	No impact of read/write split. All traffic will be sent to all servers.
Multiple Read-Write servers and multiple Read-Only servers	ScaleArc sends reads to slaves and everything else (+ some read traffic) to master (read-write server).	All traffic will be sent to all servers. **Not advised for any production traffic.**

Chapter 7: Live Monitor

Live Monitor provides real time metrics of ScaleArc health, performance and statistics on every SQL statement crossing its interfaces. Live Monitor also breaks out each cluster and centralized logging. The subsections of Live Monitor include system stats, clusterwise stats, log trail, historical stats, and diagnostics. The majority of health and performance information is provided with MRTG-like graphs.

7.1 System Stats

System Stats includes information about the ScaleArc system itself. Graphical representations for CPU Usage, Bandwidth Usage, Cache Usage, Client/Server Connections, and Queries per query type.



CPU Usage – This graph shows the real time CPU usage of the different ScaleArc processes on the ScaleArc machine.

Bandwidth Usage – This graph shows the real time network bandwidth usage in Mb/s or Gb/s, for both inbound and outbound traffic on the ScaleArc machine.

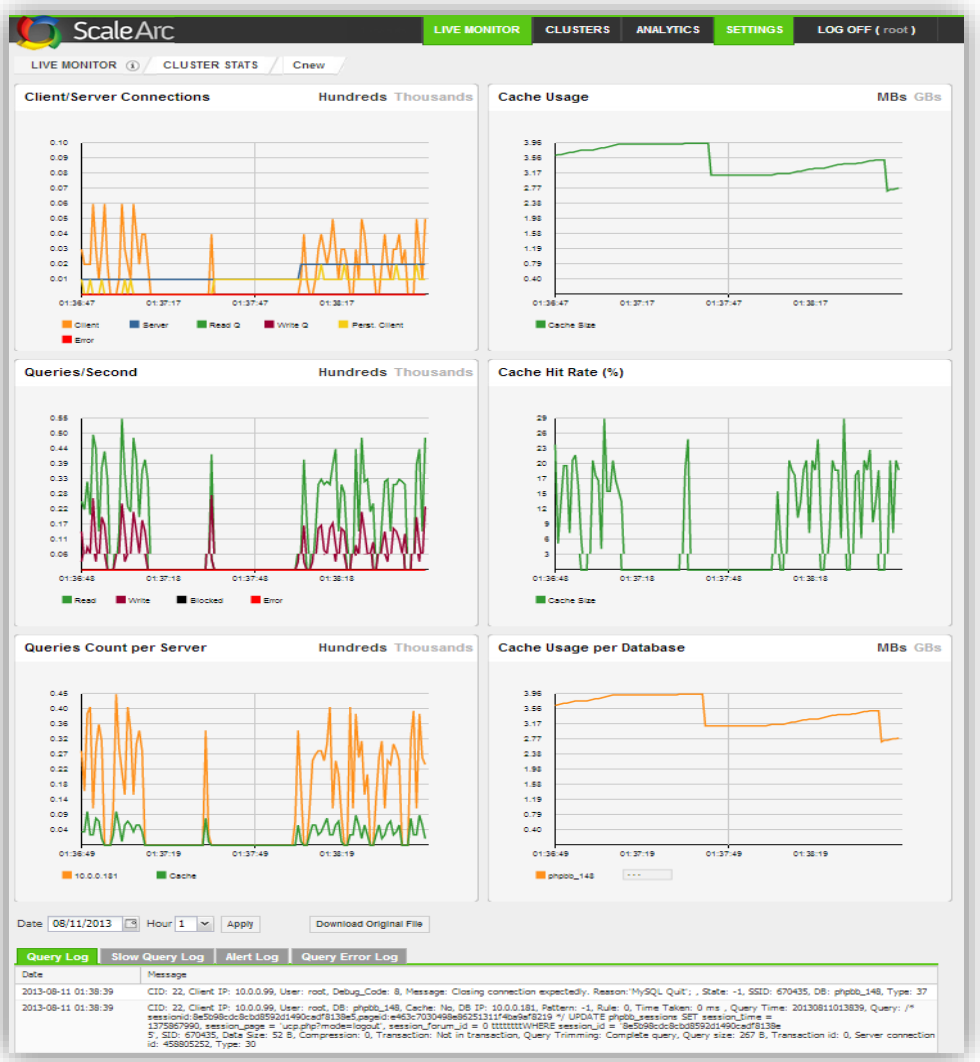
Cache Usage – This graph shows the real time memory cache usage in MBs or GBs on the ScaleArc machine.

Client/Server Connections – This graph shows the number of network connections of different types on the ScaleArc machine.

Queries – This graph shows the real time throughput of queries on the ScaleArc machine based on the different query types.

7.2 Clusterwise Stats

Clusterwise Stats includes information specific to each cluster configured in ScaleArc. Graphical representations include Client/Server Connections, Cache Usage, Queries/Second, Cache Hit Rate, and Queries Count per Server, Cache Usage per Database and Logging.



Client/Server Connections – This graph shows the number of network connections of different types on the cluster.

Cache Usage – This graph shows the real time in memory cache usage (in MBs or GBs) on the cluster.

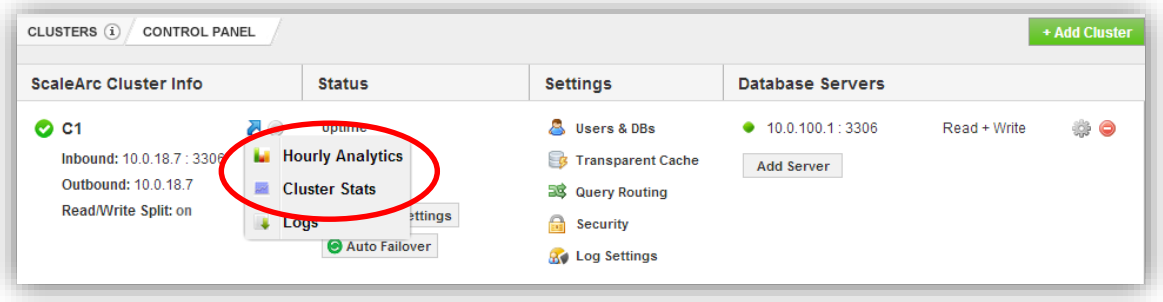
Queries/Second – This graph shows the real time throughput of queries in the clusters based on the different query types.

Cache Hit Rate – This graph shows the percentage of queries that were returned from the iDB cache of the cluster.

Queries Count/Server – This graph shows the real time count of queries for each of the database server configured on the cluster.

Cache Usage/Database – This graph shows the real time size of the cache used on iDB per database server configured on the cluster.

Live Monitor cluster stats can also be easily accessed through the quick links located next to the cluster on the ScaleArc Control Panel.



7.3 Logs

Log Trail provides a real-time view into every query and session that is established to and through ScaleArc.

Query Log – This log captures all client connections (applications) and SQL statements ran against any database server in the cluster. The query log is useful when you suspect an error from a client and want to know exactly what the client or application sent to the database server.

General Log –This log contains API status collection information regarding health of the Scalearc core.

Slow Query Log – The slow query log includes all queries which take longer than the configured time to process via your servers. You could use this log to find out which queries are most intensive, optimize them, or cache the frequently repeated ones to reduce load. Slow query logs are disabled by default. Refer to the Log Settings section for details.

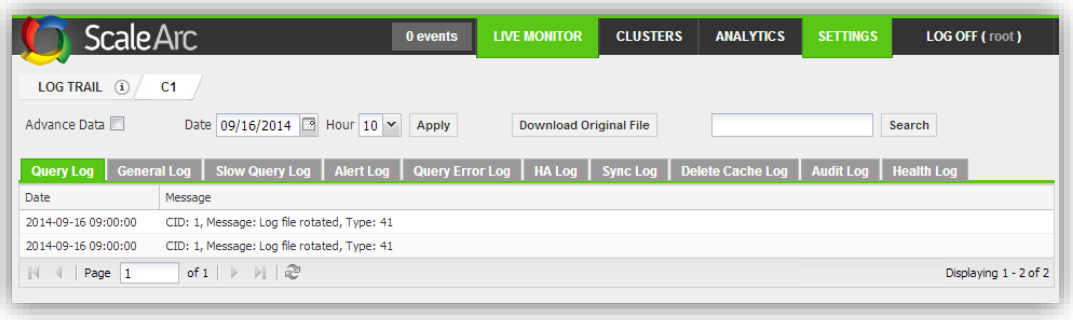
Alert Log – Alert logs include log messages from ScaleArc that requires attention of the system administrator e.g. database health check failure alerts.

Query Error Log – Error logs include log messages from ScaleArc where either there is an error processing the query or an internal error has happened.

HA Log – HA log includes log messages from ScaleArc HA service.

Sync Log – Sync log includes log messages from the rsync service that is syncing resources between the HA pair.

Delete Cache Log – This log includes messages when a cache data is deleted.



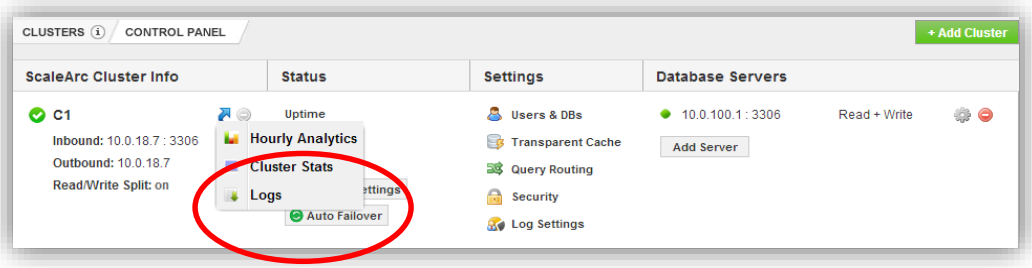
Audit Log – This log contains an audit trail for UI changes performed on ScaleArc for a given cluster. This includes which user made a change and what change was made to the system.

Health Log – This log includes messages related to the health of the database servers.

AutoFailover Log – This log includes messages related to the status of Autofailover.

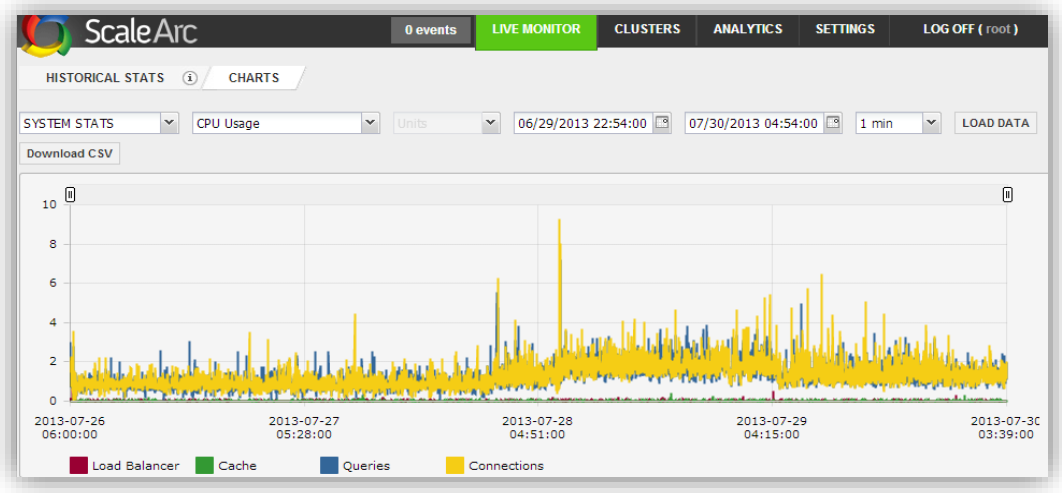
Samba Log – Contains informations pertaining to ScaleArc as a Backup Domain Controller feature.

Logs can also be easily accessed through the quick links located next to the cluster on the ScaleArc Control Panel.



7.4 Historical Stats

Historical Stats give the user a look into the system wise or cluster wise historical information of various properties. This is very useful to analyze and troubleshoot system behavior. Here are the different statistics that are displayed in the historical graphs.



System: For the ScaleArc system we display historical information for CPU Usage, Bandwidth Usage, Cache Usage, Client/Server Connections, Queries, MP Stats, and VM Stats.

Cluster: For the ScaleArc cluster we display historical information for Client/Server Connections, Cache Usage, Query/Second, Cache Hit Rate, Queries Count per Server, and Cache Usage per DB.

Provide a date range and select the resolution of the data points to view the historical graphs.

CPU usage graph - This graph shows the real time CPU usage of the system or clusters on the ScaleArc machine.

Bandwidth usage graph - This graph shows the network bandwidth usage (in MB/s or GB/s) for the system or cluster traffic on the ScaleArc machine.

Vmstats - This graph shows the virtual memory statistics of the system. See vmstat for more details.

Cache usage graph - This graph shows the cache used by the system or cluster on the ScaleArc machine.

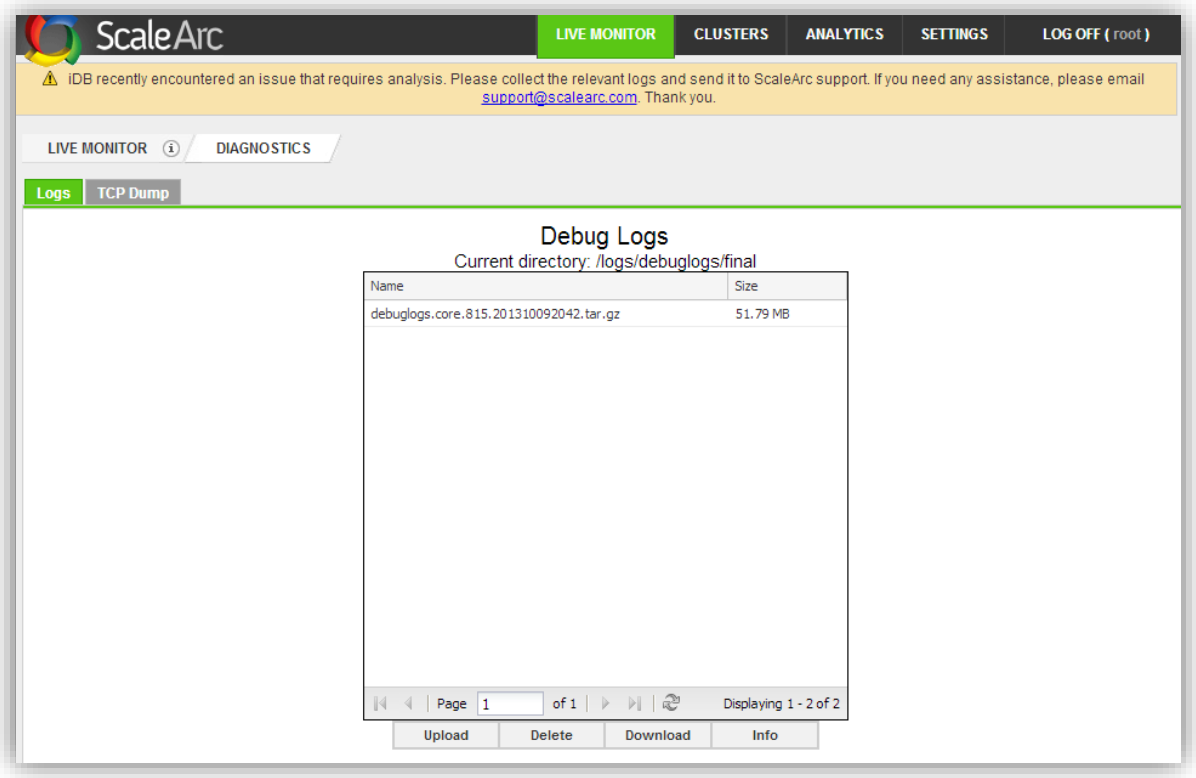
Client/Server Connections graph - This graph shows the number of network connections of different types on the system or cluster.

Query/Second graph - This graph shows the real time throughput of queries in the clusters or system.

Mpstats - This graph shows the processors related statistics. See **mpstats** for more details.

7.5 Diagnostics

The Diagnostics screen is used to view and download any debugging information from the system such as core dumps, log files, etc. needed to diagnose any system issue.

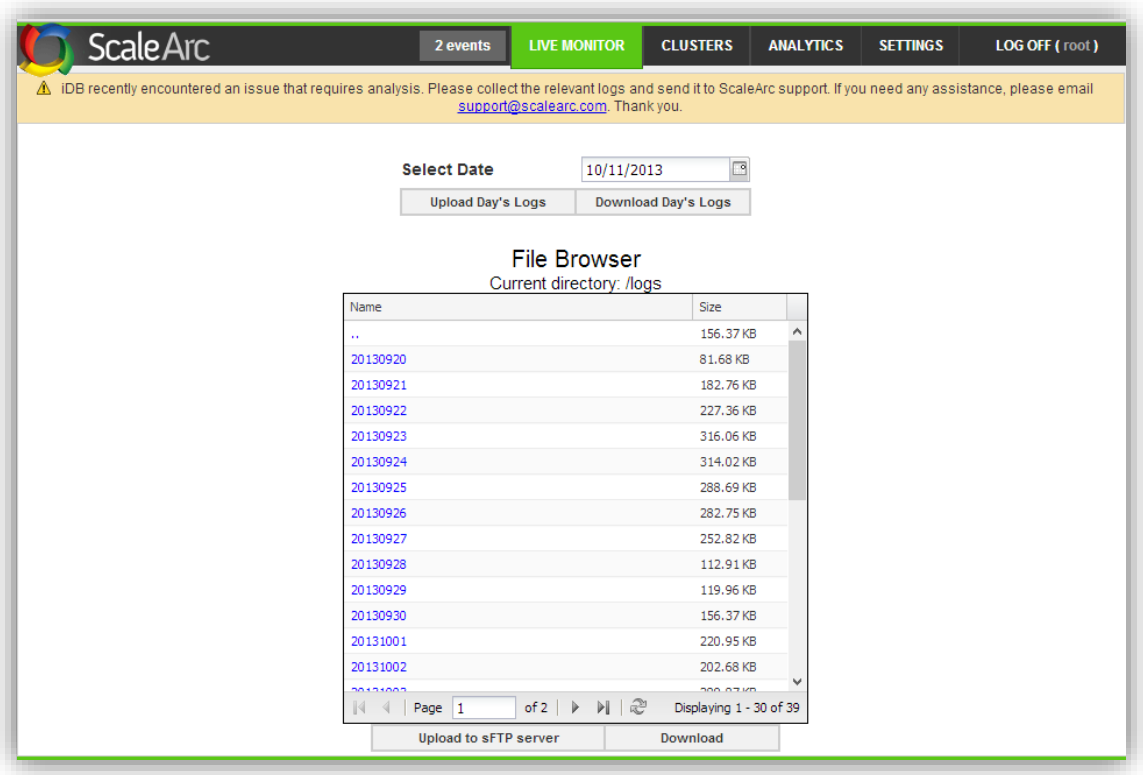


The first tab is the Logs tab in the Diagnostics screen. From here you can access the debug and daily log files.

A debug log can record database operations, system processes, and errors that occur during operation. Debug logs are a valuable tool to any system administrator. Click on the debug log file and use the options below to obtain more information, download or delete the log file. If you require assistance you can also upload the files to ScaleArc's FTP server for customer support.

To download the logs for a particular day, select the date and then click on Downloads Day's Log's.

The next screen allows you to browse through the entire ScaleArc system so that you can download/ upload a specific file for diagnosis.



The second tab is the TCP Dump tab. TCP Dumps allows you to save the packets that are captured to use for analysis.

The first step is to select the mode of capturing TCP dumps. There are three modes to select from:

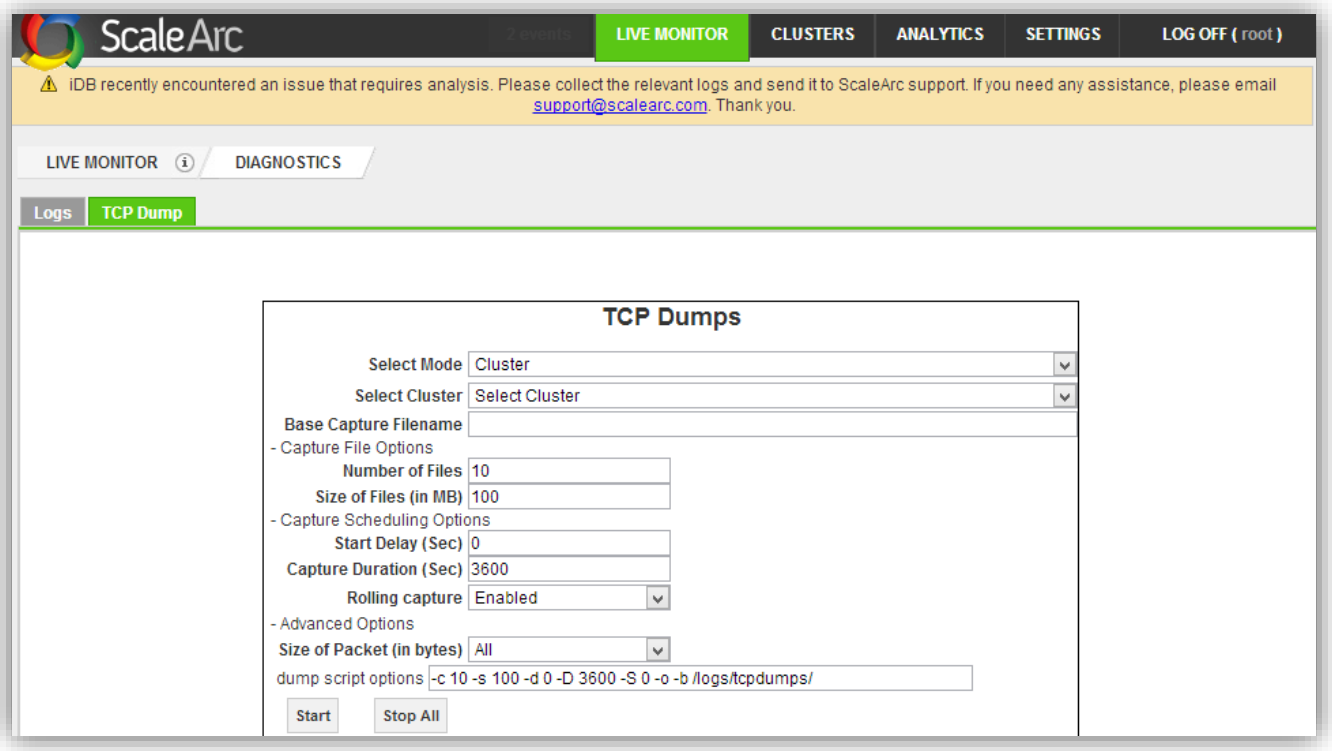
1. Cluster based TCP dumps capture data for a particular cluster.
2. Host and Port captures data for a particular IP address/hostname and port.
3. Custom mode allows you to customize a PCAP string.

After selecting the mode specify the remaining parameters.

Capture file options:

Number of files - Enter the number of files that TCP dump will create.

Size of files – From this area you can define that maximum size you would like for each file to be.



Capture scheduling option:

Start delay - Enter the number of seconds you would like for TCP dump to wait before capturing data.

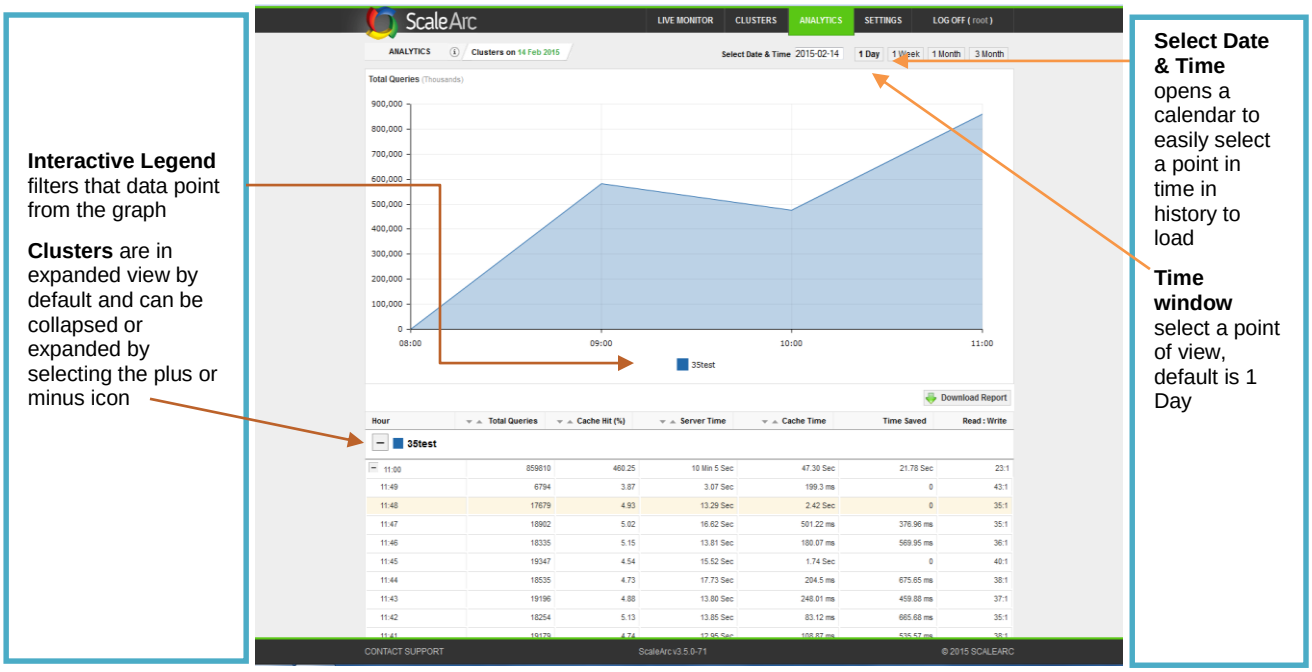
Capture duration - enter the number of seconds TCP Dump will continue to capture data before it stops.

Rolling capture - If rolling capture is enabled TCP dump will continue capturing data for the specified number of files with maximum size defined in capture file options. If disabled, TCP dump will quit when this condition is met.

Advanced options - select the amount of data that is captured in each TCP packet. Enter the dump script options and click on Start to begin capturing TCP dump files.

Chapter 8: Analytics

The Analytics of ScaleArc shows near real time view of SQL traffic and is designed to provide a simplified view of database traffic. Through each screen administrators can drill down into each SQL to see exactly which queries are taking the most resources. Analytics has an easy to use GUI which shows top consuming SQL, and has the ability to compare SQL from different dates. Additionally, SQL can be added directly to cache from within the Analytics page.



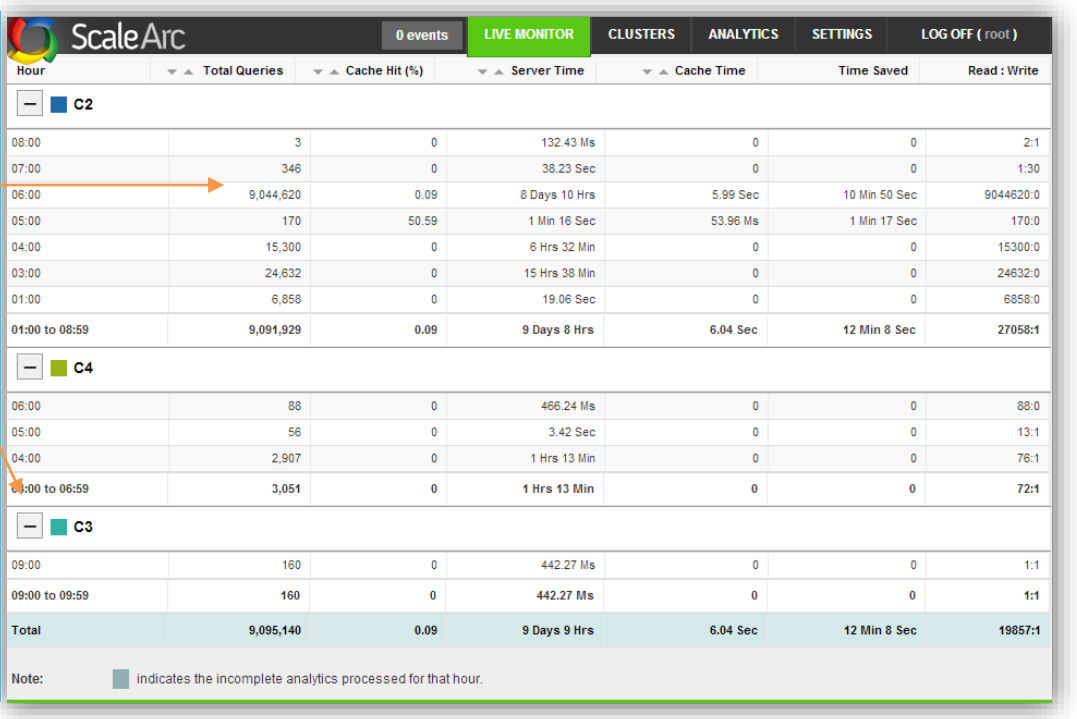
8.1 The Landing Page

The landing page of analytics breaks out information by cluster with graphs illustrating transaction trending, report style totals for the 24 hour period expandable in one hour time slices, a date field for extending the illustration to historical data and a means to modify the view from 1 day, one week, a month, 3 months or a specific date and time. Each graph within analytics provides a legend that is interactive. Click on an item icon in the legend and it filters the cluster from the graph.

Interactive Legend filters that data point from the graph

Clusters are in expanded view by default and can be collapsed or expanded by selecting the plus or minus icon

Selectable time periods drill into an hour or 24 hour time slice for more granular detail and further analysis



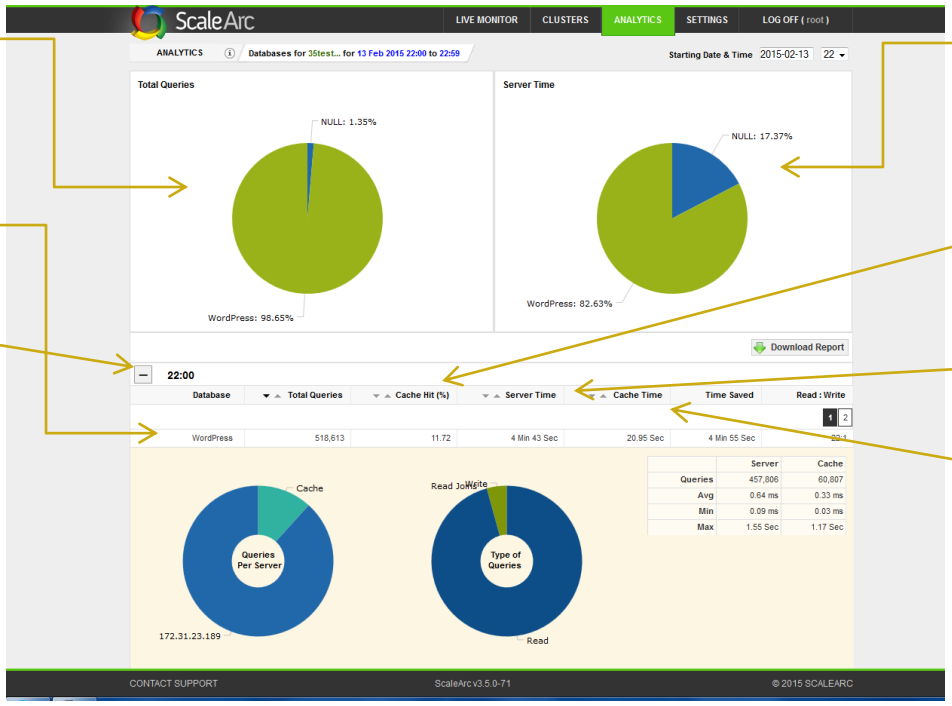
8.2 Cluster Analysis

Cluster analysis gives us a breakdown of each database associated with the selected cluster within the time frame selected.

Total Query Chart illustrates total number of queries by database by overall percentile

DB Name- list of all databases associated with cluster analysis

Hour - select detailed view in one hour time slices



Server Time Chart illustrates Server Database server process for overall query count by database.

Cache Hit (%) - cache hit percentage per database

Server Time - aggregated query DB Server process time

Cache Time, - aggregated ScaleArc cache process time

8.3 Four Quandrant Graph

In a typical production environment the four quadrant graph focuses on the majority of where SQL traffic is spending time. In the example below, ScaleArc analytics focuses on the Frequent/Infrequent, Fast/Slow Query Graph and denotes balloons with hover actions illustrating exactly which SQL queries each represents.

Frequent Fast Query-- High volume low impact on process, these are the best performing queries

Infrequent Fast Query - high performance less frequent queries

X Axis = frequency

Y Axis = transaction time to completion

Frequent Slow Query - is the highest cost query in terms of response time and server process. This identifies potential "problem queries".

Infrequent Slow Queries - high cost but not as frequent. This quadrant also identifies "problem queries".

8.4 Database Analysis

Database Analysis - provides SQL structures and performance analysis

Balloon - random color coding for distinctiveness each balloon corresponds to a SQL structure listed below in the tabular report and when selected highlights the corresponding line in the report below

Starting Date - select from the archive of historical data from the ScaleArc storage

Balloon Click - highlights the line in the report that references the same the query structure

Balloon hover - provides an illustration of the SQL Query Structure

By selecting the balloon, the corresponding line in the report is highlighted providing a more detailed result of performance both from the database and the ScaleArc cache. You can also filter specific query types. For example, you can select to view only the reads query types by checking the reads check box.

8.5 Comparative Analysis

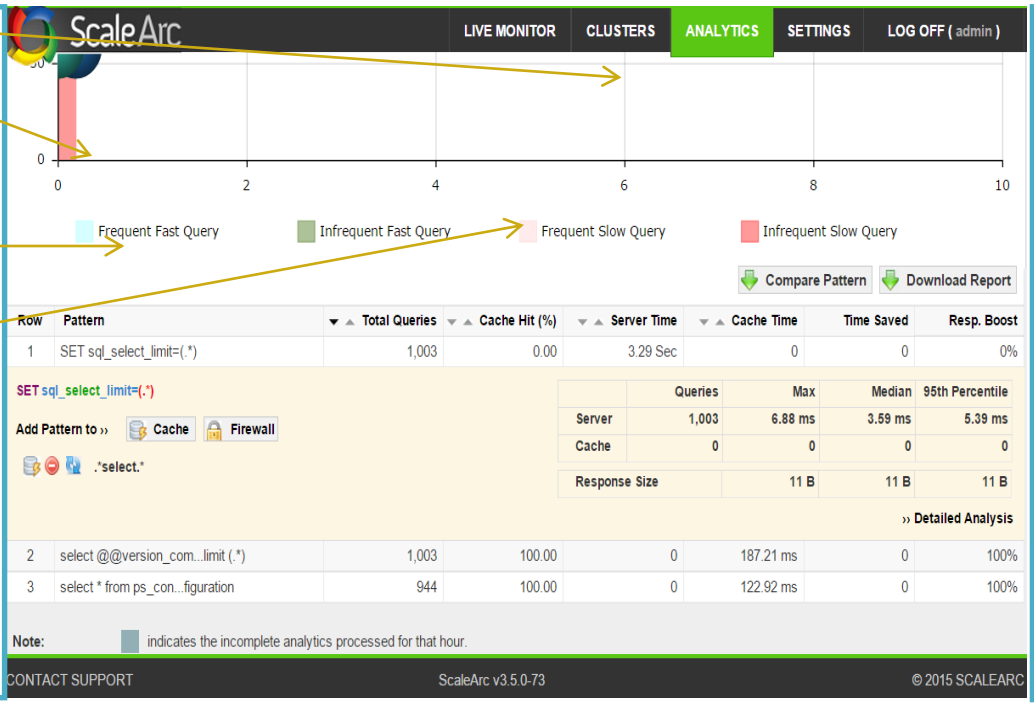
Comparative analysis allows administrators and developers to understand changes in SQL patterns and utilization.

Compare Patterns

Query Click - hyperlinks to the SQL statement analysis report

Add – adds **to** cache

Query hover - provides an illustration of the SQL Query Structure



Total Queries – total number of queries for the pattern in the hour

Cache Hit (%) – The % of total queries returned from the ScaleArc cache

Server Time – the total amount of time taken by the server to respond to queries

Cache Time – the total time taken by ScaleArc to respond from cache

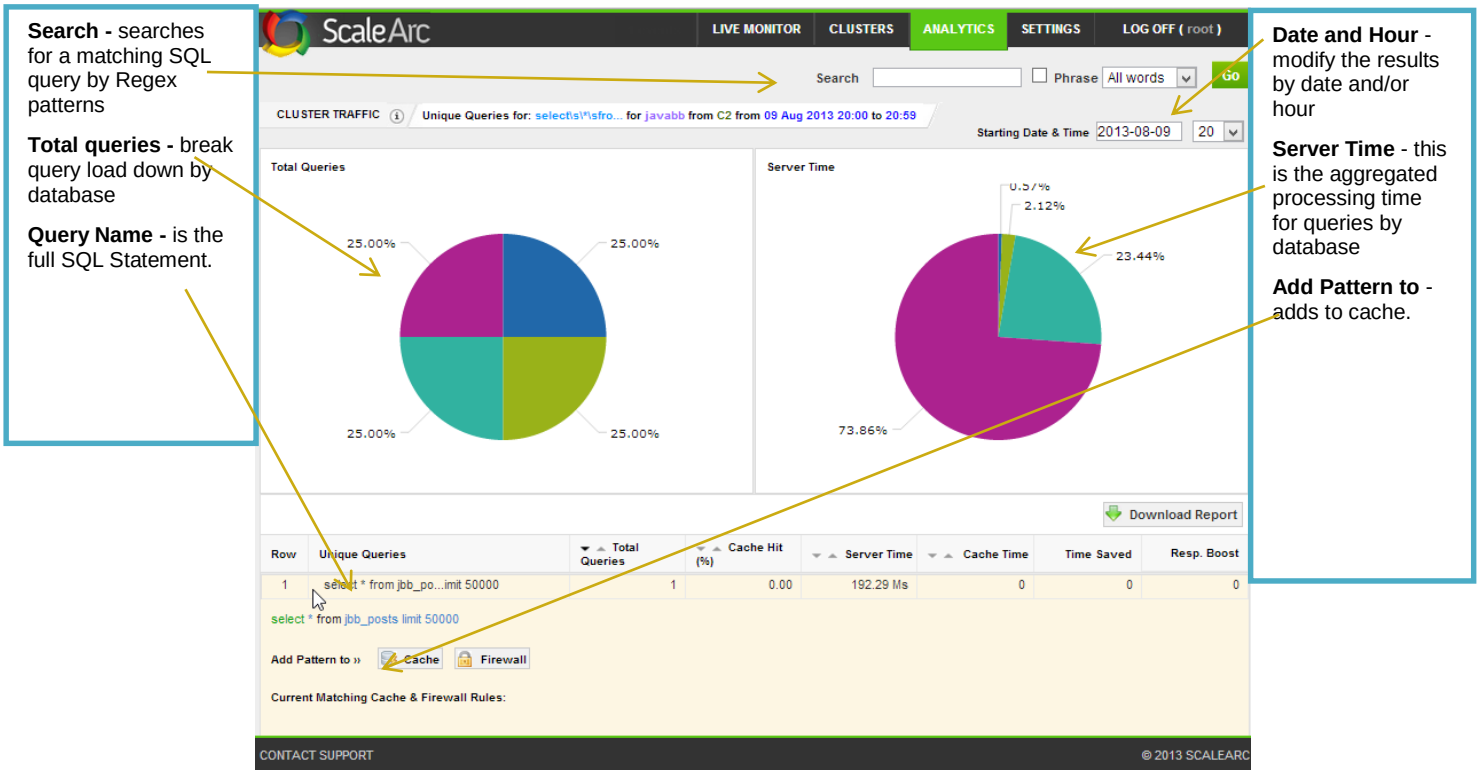
Time Saved – total time saved by ScaleArc cache

8.6 SQL Structure

SQL structure is the SQL Query with its variables masked. This allows for easy SQL query grouping for Analytics to report and easy to identify problem statements. Each SQL structure is actionable, so adding cache is simple and flexible without having to create rules for each query or figuring out the Regular Expressions necessary to match the pattern of your SQL statement. Clicking on the query structure in the tabular report links to a unique query report for further analysis.

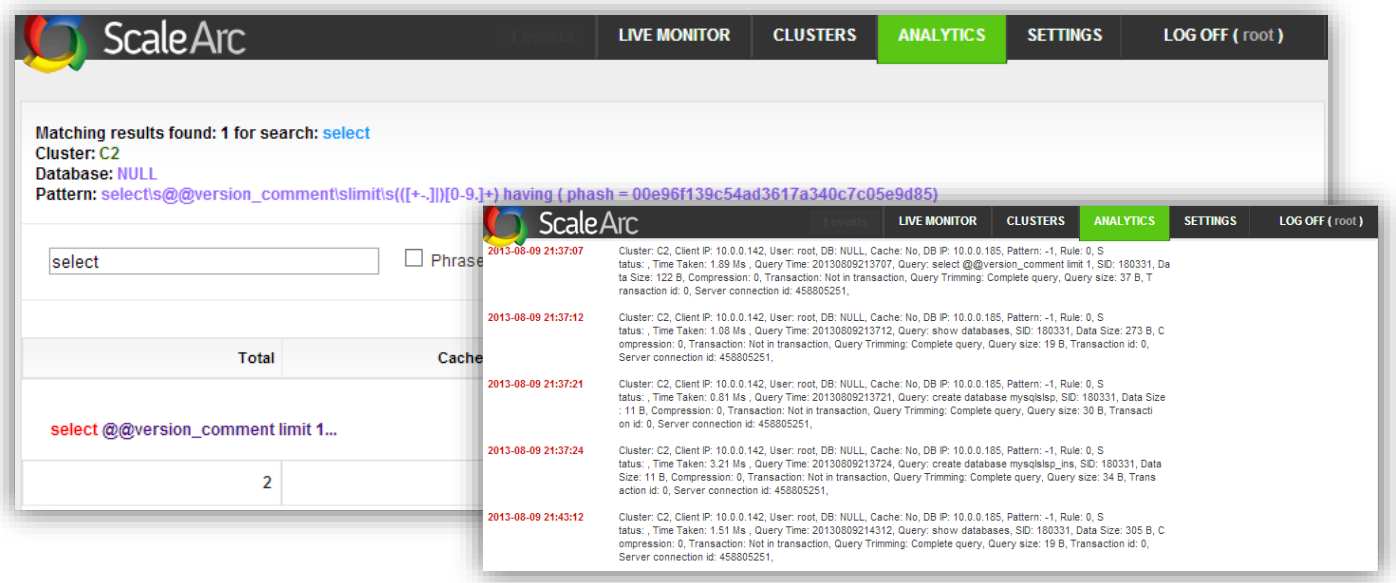
8.6.1 SQL Statement Analysis

SQL statement analysis is a detailed account for the selected SQL structure per database. The SQL statement analysis gives us unique information relative to each query falling under the selected SQL statement structure.



8.7 Log Search

Log Search will match text patterns to all SQL queries within the time frame specified. If found, the result will display the query with statistics. The query itself is hyperlinked and will list every log entry details.



Appendix A: ScaleArc Deployment Checklist

Please fill in the following details before you start the deployment/configuration process.

*Note: Fields in bold font marked with * are mandatory.*

1. Network Configuration:

This is the basic configuration needs to be done to access primary ScaleArc appliance in your network.

Field Name	Value	Comments
+***Hostname**	**Primary**	Default value: ScaleArcha1, valid characters: a-z 0-9 _ - .
	Secondary	
***Interface**	**Primary**	Default: eth0, in case of multiple network cards, you can select any one of them.
	Secondary	
***IP Address**	**Primary**	Default value: 192.168.1.10, using this you can access ScaleArc UI. Also referred as Management IP.
	Secondary	
***Subnet**	**Primary**	Default value: 255.255.255.0
	Secondary	
Gateway		
Primary DNS		
Secondary DNS		
Time Zone		Default value: (GMT -12:00) Eniwetok, Kwajalein

Table 1

+Make sure you specify different Host Name and IP Address for each ScaleArc appliance if you are planning to configure in HA. For HA the Secondary ScaleArc appliance has to be configured the same as Primary ScaleArc with the same timezone and user access credentials

2. ScaleArc User Access Credentials:

By default, ScaleArc has single user, admin, which can be used to make changes to the configuration. Same user, with monitoring password can be used to monitor the live stats, but won't be able to make any changes. For the 1st time log in to ScaleArc, you can use admin user with admin as password to start the configuration.

Field Name	Value	Comments
***User Name**		Default value: admin
***Admin Password**		Default value: admin@123
***Monitoring Password**		

Table 2

Note: When you configure HA, username and password(s) of primary ScaleArc appliance will get overwritten on to secondary ScaleArc appliance.

3. HA configuration:

This step is required only when you want to configure HA between 2 ScaleArc appliances. DATE/TIME should be synced on both the ScaleArc appliances before configuring HA. HA VIP will be configured on the same network interface on which individual Management IP is configured.

Field Name	Value	Comments
*Floating Virtual IP		VIP will always point to Primary ScaleArc server once HA is configured. Also referred as HA Management IP.

Table 3

4. Configuration of extra network interface(s):

Note: If you want to configure bonding link aggregation, please contact ScaleArc sales team.

ScaleArc appliance comes with atleast 5 network interfaces. The number of network interfaces may change based on ScaleArc 1000, ScaleArc 200, ScaleArc 4000 or if you are using custom hardware / VM. You can use any one or multiple network interfaces to configure cluster. Eth0 is used as management / default interface & same can also be used for application cluster. If you are planning to use any network interface mentioned below, make sure you change the IP Address & Subnet for the same.

Network Interface	IP Address & Subnet	Comments
Eth1	IP	Default IP: 192.168.1.11, Subnet: 255.255.255.0
	Subnet	
Eth2	IP	Default IP: 192.168.1.12, Subnet: 255.255.255.0
	Subnet	
Eth3	IP	Default IP: 192.168.1.13, Subnet: 255.255.255.0
	Subnet	
Eth4	IP	Default IP: 192.168.1.14, Subnet: 255.255.255.0
	Subnet	

Table 4

5. Cluster Configuration:

To create a cluster, you need at least one IP address. IP address can either be a physical IP address or floating VIP. In case of HA the IP has to be a floating one. Inbound IP address & Port will be used by the application to connect to ScaleArc cluster, while Outbound IP address will be used by ScaleArc server to connect to backend database server(s) in that cluster. Inbound & Outbound IP address can be on same or different network interfaces.

Field Name	Value	Comments
***Cluster Name**		
***Instance Name**		**Applicable only for MS SQL.** Default MSSQLSERVER.
***Inbound Network Interface**		Any network interface on which physical IP address is configured.

***Inbound IP Address**		You can use existing physical IP address or create a new floating VIP.
***Inbound Subnet**		This is required only when you want to create new VIP for the cluster.
***Cluster Port**		Default value: 1433 (MS SQL)
***Outbound Network Interface**		This can be same as Inbound NIC or different.
***Outbound IP Address**		This can be same as Inbound IP address or any other existing VIP or create a new floating VIP.
***Outbound Subnet**		This is required only when you want to create new VIP for the cluster.

Table 5

If you want to create multiple clusters, please use the copy of Table 5.

6. Database User Credentials:

For ScaleArc to be able to access database server(s), you should provide database user credentials & necessary permissions for the cluster outbound IP address & management IP address. This information will be used by ScaleArc to monitor database server health & slave replication lag time.

Field Name	Value	Comments
***User Name**		Database server user name.
***Password**		Password for the above user.

Table 6

7. Database Server(s):

Specify the database server details which you will be adding to one or multiple clusters.

Field Name	Value	Comments
***DB Server IP**		IP address of database server.
***Port**		Default value: 1433 (MS SQL)
***Server Type**		Read/Write (Master) or Read Only (Slave)
Version		Database server version, eg. MS SQL Server 2008 R2, etc.

Table 7

If there are multiple database servers, please use the copy of Table 7.

Questionnaire:

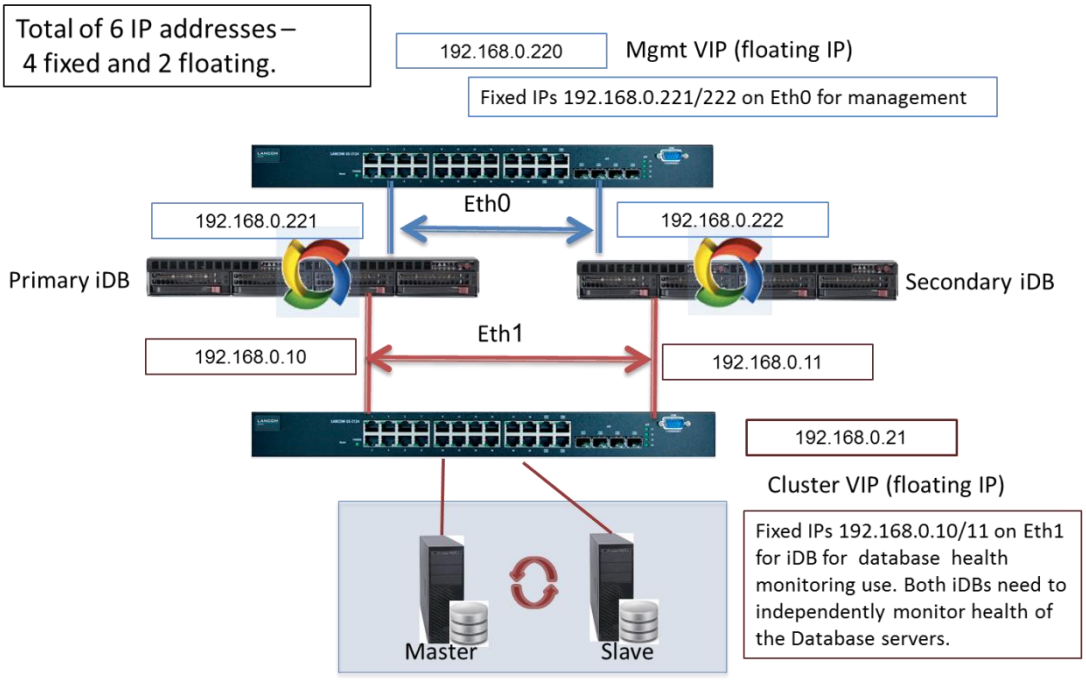
Question	Answer
Are the application server(s) & database server(s) at the same location? This will help us to identify if there can be any latency.	
What is the platform & version for application server(s), Linux / MS Windows / UNIX, etc.?	

What is the platform & version for database server(s), Linux / MS Windows / UNIX, etc.?	
Are you using client side connection pooling from application?	
If there are multiple database servers, then, what type of replication is configured?	
Are you using any in memory data caching technology like memcached?	
ScaleArc deployment platform, appliance / VM / Xen / Custom HW / etc.?	

Table 9

Sample IP address Configuration in One-Arm mode for iDB HA Pair

Applies for all (VM/MySQL/MSSQL) Deployments



Appendix B: ScaleArc Third-Party Software Dependencies

ScaleArc ships with a large collection of third-party software that performs a wide variety of functions.

A large portion of this software is distributed under various open-source license agreements. ScaleArc includes copies of all applicable open-source licenses on your server, and ensures that source RPMs are publicly available.

The following is a list of third-party software that ScaleArc manages:

Open Source

Module	3rd Party Software	License	Copyright Information
Core	glibc-2.12-1.149.el6_6.9.x86_64	LGPL	This product includes software developed by Copyright (C) 2007 Free Software Foundation, Inc. http://fsf.org/. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
Core	libstdc++-4.4.7-11.el6.x86_64	GPLv3	This product includes software developed by Copyright (C) 2009 Free Software Foundation, Inc.. Permissions and limitations regarding this software are subject to terms available at http://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
Core	libselinux-2.0.94-5.8.el6.x86_64	Public Domain Software	This library (libselinux) is public domain software. Permissions and limitations regarding this software are subject to terms available at: http://userspace.selinuxproject.org/trac/browser/libselinux/LICENSE
Core	libcom_err-1.41.12-21.el6.x86_64	MIT	This product includes software copyright © 1985-2014 by the Massachusetts Institute of Technology. Permissions and limitations regarding this software are subject to terms available at http://www.rpmfind.net/linux/RPM/centos/6.5/i386/Packages/libcom_err-1.41.12-18.el6.i686.html
Core	libgcc-4.4.7-11.el6.x86_64	GPLv2	This product includes software developed by Copyright (C) 2009 Free Software Foundation, Inc.. Permissions and limitations regarding this software are subject to terms available at http://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
Core	krb5-libs-1.10.3-37.el6_6.x86_64	MIT	This product includes software copyright © 1985-2014 by the Massachusetts Institute of Technology. Permissions and limitations regarding this software are subject to terms available at http://web.mit.edu/~kerberos/krb5-devel/doc/mitK5license.html
Core	keyutils-libs-1.4-5.el6.x86_64	GPL/LGPL	This product includes software Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>. Permissions and limitations regarding this software are subject to terms available at http://people.redhat.com/~dhowells/keyutils/keyutils-1.1-4/keyutils.spec
Core	pcre-7.8-6.el6.x86_64	BSD	This product includes software copyright © 1997-2007 University of Cambridge. Permissions and limitations regarding this software are subject to terms available at http://www.pcre.org/license.txt.

Module	3rd Party Software	License	Copyright Information
Core	zlib-1.2.3-29.el6.x86_64	Zlib	Interface of the 'zlib' general purpose compression library version 1.2.3, July 18th, 2005 Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler. All rights reserved and redistribution of this software is subject to terms available at http://pkgs.org/centos-6/centos-i386/zlib-1.2.3-29.el6.i686.rpm.html
Core	openssl-1.0.1e-30.el6.11.x86_64	OpenSSL	This product includes software developed by theOpenSSLProject for use in theOpenSSLToolkit(copyrightTheOpenSSLProject.All Rights Reserved) and redistribution of this software is subject to terms available at http://www.openssl.org and http://www.openssl.org/source/license.html.
Core, UI, Services	sqlite-3.6.20-1.el6_7.2.x86_64	Public domain	License: Public domain. All rights reserved and redistribution of this software is subject to terms available at http://sqlite.org/copyright.html.
Services	python-libs-2.6.6-52.el6.x86_64	Python Software Foundation v2	Copyright © 201- 2012 Python Software Foundation; All Rights Reserved and redistribution of this software is subject to terms available at http://www.python.org/download/releases/2.2/license/
Services	pyodbc-3.0.7-1.x86_64.rpm	MIT	This product includes software copyright © 1985-2014 by the Massachusetts Institute of Technology. Permissions and limitations regarding this software are subject to terms available at http://opensource.org/licenses/MIT
Services	MySQL-python-1.2.3-0.3.c1.1.scalearc.x86_64	GPLv2+	This product includes software Copyright (C) 1989, 1991 Free Software Foundation, Inc., Permissions and limitations regarding this software are subject to terms available at http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
Services	unixODBC-2.2.14-14.el6.x86_64	GPL/LGPL	This product includes software Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>. Permissions and limitations regarding this software are subject to terms available at http://people.redhat.com/~dhowells/keyutils/keyutils-1.1-4/keyutils.spec
Services	psutil-3.1.1-1.x86_64	BSD	Copyright (c) All rights reserved and redistribution of this software is subject to terms available at http://www.rpm-find.net/linux/RPM/ftp.scientificlinux.org/linux/scientific/6.0/x86_64/os/Packages/pcre-7.8-3.1.el6.i686.html
Services	python-crontab-1.4.3-1.noarch.rpm	GNU	Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>
Services	python-dateutil-1.4.1-6.el6.noarch.rpm	PSF	Python License: **http://opensource.org/licenses/Python-2.0**
Services	ipaddr-2.1.10-1.noarch.rpm	Apache	Apache License: **http://www.apache.org/licenses/**LICENSE-2.0
UI, Services	freetds-0.92-4.x86_64	LGPL	This product includes software copyright © 1998-2008 Brian Bruns et al. and is licensed under the GNU LGPL, version 2 or later. Permissions and limitations regarding this software are subject to terms available at http://www.freetds.org/faq.html#license

Module	3rd Party Software	License	Copyright Information
UI, Services	dmidecode-2.12-5.el6_6.1.x86_64	GPLv2	This product includes software Copyright (C) 1989, 1991 Free Software Foundation, Inc., Permissions and limitations regarding this software are subject to terms available at http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
UI, Tools	Java 1.7.0	Java License Agreement	This product includes for the Java SE Platform Products and JavaFX. Permissions and limitations regarding this software are subject to terms of Oracle Binary Code License Agreement available at http://www.oracle.com/technetwork/java/javase/terms/license/index.html.
UI, API, Services	curl-7.19.7-40.el6_6.4.x86_64 libcurl-7.19.7-40.el6_6.4.x86_64 python-pycurl-7.19.0-8.el6.x86_64	Curl	This product includes Curl software which is Copyright 1996-2007, DanielStenberg, <daniel@haxx.se>.All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http://curl.haxx.se/docs/copyright.html.
UI	Jquery 1.4.2	Jquery	Copyright © 2011 jQuery Team. All rights reserved. Permissions and limitations regarding this software are subject to MIT license terms avaialble at http://opensource.org/licenses/MIT
UI	bzip - bzip2-1.0.5-7.el6_0.x86_64 bzip2-libs-1.0.5-7.el6_0.x86_64	bzip2	Copyright © 1996-2005 Julian R Seward This file is a part of bzip2 and/or libbzip2, a program and library for lossless, block-sorting data compression. Copyright © 1996-2005 Julian R Seward. All rights reserved. Permissions and limitations regarding this software are subject to terms available at http://www.bzip.org.
Services	snmp - net-snmp-libs-5.5-54.el6_7.1.x86_64 net-snmp-5.5-44.el6_4.4.x86_64 net-snmp-utils-5.5-54.el6_7.1.x86_64	SNMP	Various copyrights apply to this package. Permissions and limitations regarding this software are subject to terms available at http://www.net-snmp.org/about/license.html.
Services	redis-2.8.11-1.el6.remi.x86_64.rpm python-redis-2.10.1-1.noarch.rpm	BSD	Copyright (c) All rights reserved and redistribution of this software is subject to terms available at http://redis.io/topics/license
Services	python-redis-2.10.1-1.noarch.rpm	MIT	This product includes software copyright © 1985-2014 by the Massachusetts Institute of Technology. Permissions and limitations regarding this software are subject to terms available at http://opensource.org/licenses/MIT
Services	MySQL-python-1.2.3-0.3.c1.1.scalearc.x86_64	GPLv2+	This product includes software Copyright (C) 1989, 1991 Free Software Foundation, Inc., Permissions and limitations regarding this software are subject to terms available at http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
Services	gevent-1.0.1-1.x86_64.rpm	MIT	This product includes software copyright © 1985-2014 by the Massachusetts Institute of Technology. Permissions and limitations regarding this software are subject to terms available at http://opensource.org/licenses/MIT

Module	3rd Party Software	License	Copyright Information
Services	samba-4.1.17-1.el6.x86_64	GPL	This product includes software Copyright (C) 1989, 1991 Free Software Foundation, Inc., Permissions and limitations regarding this software are subject to terms available at **http://www.samba.org/samba/docs/GPL.html**
Services	krb5-appl-servers-1.0.1-7.el6_2.1.x86_64.rpm	MIT	This product includes software copyright © 1985-2014 by the Massachusetts Institute of Technology. Permissions and limitations regarding this software are subject to terms available at http://web.mit.edu/~kerberos/krb5-devel/doc/mitK5license.html
Services	krb5-appl-clients-1.0.1-7.el6_2.1.x86_64.rpm	MIT	This product includes software copyright © 1985-2014 by the Massachusetts Institute of Technology. Permissions and limitations regarding this software are subject to terms available at http://web.mit.edu/~kerberos/krb5-devel/doc/mitK5license.html
Services	krb5-workstation-1.10.3-37.el6_6.x86_64	MIT	This product includes software copyright © 1985-2014 by the Massachusetts Institute of Technology. Permissions and limitations regarding this software are subject to terms available at http://web.mit.edu/~kerberos/krb5-devel/doc/mitK5license.html
Services	krb5-server-1.10.3-37.el6_6.x86_64	MIT	This product includes software copyright © 1985-2014 by the Massachusetts Institute of Technology. Permissions and limitations regarding this software are subject to terms available at http://web.mit.edu/~kerberos/krb5-devel/doc/mitK5license.html
Services	cx_Oracle-5.1.1-1.x86_64.rpm	PSF	Python License: **http://opensource.org/licenses/Python-2.0**
Services	oracle-instantclient11.2-basic-11.2.0.4.0-1.x86_64.rpm	OTN	**Oracle Technology Network Developer License: http://www.oracle.com/technetwork/licenses/standard-license-152015.html**
Services	djangorestframework-2.3.14-1.noarch.rpm	BSD	Copyright (c) All rights reserved and redistribution of this software is subject to terms available at http://www.django-rest-framework.org/#license
Services	Django-1.6.5-1.noarch.rpm	BSD	**Copyright (c) Django Software Foundation and individual contributors. All rights reserved and** redistribution of this software is subject to terms available at https://github.com/django/django/blob/master/LICENSE
Installer	puppet-3.7.3-1.el6.noarch.rpm	Apache 2.0	Apache license: http://www.apache.org/licenses/LICENSE-2.0.html
Installer	ruby-1.8.7.374-4.el6_6.x86_64.rpm	Ruby	**Ruby license: https://www.ruby-lang.org/en/about/license.txt**
Services	mysql-connector-python-2.0.1-1.el6.noarch.rpm	GPL	This product includes software Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>.
Services	mysql-utilities-1.5.2-1.el6.noarch.rpm	GPL	This product includes software Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>.
Services,UI	rsync-3.0.9-1.el6.rfx.x86_64.rpm	GPL	This product includes software Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>.

Module	3rd Party Software	License	Copyright Information
Core OpenSSH	openssh-5.3p1-112.el6_7.x86_64 openssh-server-5.3p1-112.el6_7.x86_64 openssh-clients-5.3p1-112.el6_7.x86_64	BSD	BSD license: **http://www.openssh.com/**
Core	nss-3.19.1-5.el6_7.x86_64 nss-tools-3.19.1-5.el6_7.x86_64 nss-softokn-freebl-3.14.3-23.el6_7.x86_64 nss-sysinit-3.19.1-5.el6_7.x86_64 nss-softokn-3.14.3-23.el6_7.x86_64 nss-util-3.19.1-2.el6_7.x86_64	MPL 2.0	Mozilla Public License: **https://www.mozilla.org/en-US/MPL/2.0/**
Core	ntp-4.2.6p5-5.el6.centos.2.x86_64 ntpdate-4.2.6p5-5.el6.centos.2.x86_64	MIT and BSD, BSD with advertising and GPLv2	NTP copyright information: **https://www.eecis.udel.edu/~mills/ntp/html/copyright.html**
Core	nspr-4.10.8-2.el6_7.x86_64	MPL 2.0	Mozilla Public License: **https://www.mozilla.org/en-US/MPL/2.0/**
Core	libpng-1.2.49-2.el6_7.x86_64.rpm	zlib	zlib license: **http://www.zlib.net/zlib_license.html**
Core	java-1.7.0-openjdk-1.7.0.91-2.6.2.2.el6_7.x86_64.rpm	ASL 1.1, ASL 2.0, GPL+, GPLv2 and GPLv2 with exceptions and LGPL+, LGPLv2, MPLv1.0, MPLv1.1, Public Domain and W3C	OpenJDK license: **http://openjdk.java.net/bylaws#_A**
Core	libuser-0.56.13-8.el6_7.x86_64	GPLv2+	This product includes software Copyright (C) 1989, 1991 Free Software Foundation, Inc., Permissions and limitations regarding this software are subject to terms available at http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt

Self Compiled Libraries

Module	Third-Party software	License
UI, Services	mod_wsgi-3.4	Apache License
UI	Apache	Apache License
Core	sysbench-0.5-6.scalearc.1.x86_64	GPLv2

Proprietary software

Module	Third-Party software	License
Core	quicklz	Proprietary (purchased)

Appendix C: List of Ports on ScaleArc

Open Ports on Scalearc:

Default
---
80 - HTTP
443 - HTTPS
161 - SNMP

Configurable Ports:

While configuring a cluster, ScaleArc listens on the ports configured for application/web servers to use for database traffic.

Reserved Ports:

4000 - 5010 - Used by Scalearc internally for its own communication