# Cloud, Big Data and Mobile

Thursday, September 5, 2013

## Exploring Amazon RDS MySQL Second Tier Read Replica

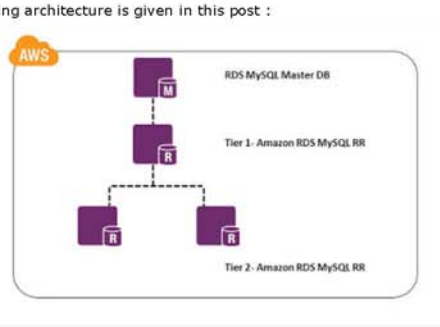
AWS recently introduced Second Tier Replica for RDS MySQL.You can use this feature to shift the load from primary master DB to the replica in first tier and prepare your application for handling extremely high level of read traffic. In this post we are going to explore this feature in following aspects :

- Steps to configure Multi-Tiered Amazon RDS MySQL Read replicas
- Second Tier Read Replica Deployment architectures
- Important Points to note

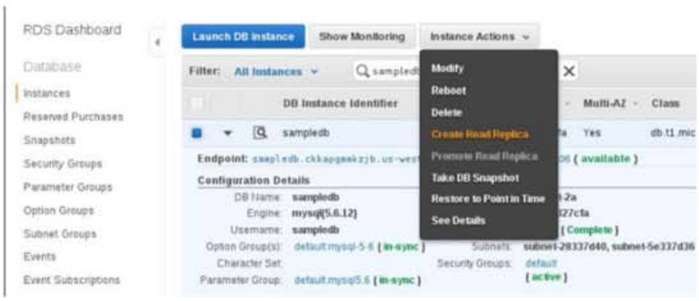
Amazon RDS MySQL version 5.6.12 on AWS WEST region was used.

### Steps to configure Multi- Tiered Amazon RDS MySQL Read Replicas

Configuration steps for the following architecture is given in this post :

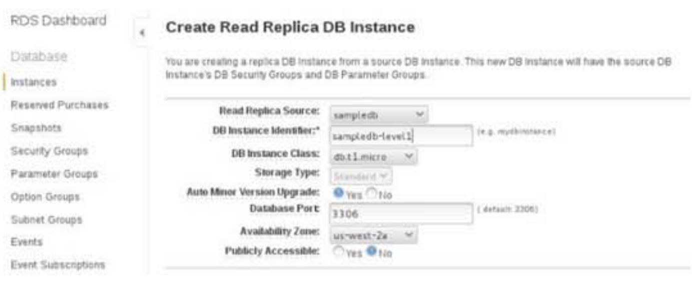


**Step 1: Creating Read Replica from Master and Place at Tier 1:**
To create RDS MySQL Read replica navigate to the dashboard of Amazon RDS, select the Amazon RDS MySQL Master(named **"sampledb"**) and use the option of "Create Read Replica".



Filter: All Instances
DB Instance Identifier | Multi-AZ | Class
sampledb | Yes | db.t1.mic
Endpoint: sampledb.ckkapgmkzjb.us-west ( available )
Configuration Details
DB Name: sampledb
Engine: mysql(5.6.12)
Username: sampledb
Option Group(s): default:mysql-5-6 ( in-sync )
Character Set:
Parameter Group: default.mysql5.6 ( in-sync )
( Complete )
Subnets: subnet-28337d40, subnet-5e337d36
Security Groups: default ( active )

Name the newly created Amazon RDS Read replica as **"sampledb-level1"** and place it in Tier 1. The Tier 1 Amazon RDS MySQL read Replica can be created in same AZ of Master or in a different AZ for High Availability. When the Tier replica is placed in Different AZ , you should factor few extra milliseconds of latency during replication.



Create Read Replica DB Instance

You are creating a replica DB Instance from a source DB Instance. This new DB Instance will have the source DB Instance's DB Security Groups and DB Parameter Groups.

Read Replica Source: sampledb
DB Instance Identifier:* sampledb-level1 (e.g. mydbinstance)
DB Instance Class: db.t1.micro
Storage Type: Standard
Auto Minor Version Upgrade: ● Yes ○ No
Database Port: 3306 ( default: 3306)
Availability Zone: us-west-2a
Publicly Accessible: ○ Yes ● No

**Explore the status from Master DB:** Post successful creation of the Tier 1 Read Replica, You can see the Read Replica Id's from exploring the details of the Master DB. Illustrated in Below Screen shot :

**Need consulting help ? Email me @**
harish11g.aws@gmail.com
Connect @ Twitter / Linkedin

**Search**

Search

**Popular Posts - Week**

25 Best Practice Tips for architecting your Amazon VPC

Dissecting Amazon ELB : 18 things you should know

Overcoming Outages in AWS : High Availability Architectures

Caching architectures using Memcached & Amazon ElastiCache

ElastiCache Redis : How to Backup & Launch from RDB Snapshots ?

Comparison Analysis:Amazon ELB vs HAProxy EC2

Load Balancing Amazon RDS Read Replica's using HAProxy

Exploring Amazon Availability Zones

Part 1: Understanding Amazon Elastic Block Store

Part 7: AWS High Availability Patterns : Multi Region Architecture

**Tags**

AWS (161) Amazon web services (103) Amazon EC2 (102) Cloud computing (97) Architecture (66) High Availability (45) Availability Zones (32) BigData (29) Tips (29) Configuration (27) Amazon S3 (25) Amazon VPC (25) AWS Region (23) Amazon Elastic Load Balancing (23) Caching (23) EC2 (22) Amazon ELB (21) EBS (20) Performance (19) Cost Savings (18) CloudSearch (17) RDS (17) elasticache (17) Apache Solr (16) Load Balancing (16) Auto Scaling (15) ELB (15) Replication (15) Mobile (14) Search (14) Amazon SQS (13) Memcached (13) autoscaling (13) Amazon AutoScaling (12) Solr (12) Amazon CloudSearch (11) Amazon CloudWatch (11) Amazon DynamoDB (11) Comparison (11) HAProxy (11) Scaling (11) Elastic IP (10) ElasticCache (10) Amazon Glacier (9) ecommerce (9) Cloudfront (8) Hadoop (8) Akamai (7) MySQL (7) Route53 (7) Sharding (7) Varnish (7) EBS-Optimized (6) Geospatial (6) MapReduce (6) PIOPS (6) Apache SolrCloud (5) DR (5) LBR (5) Multi AZ (5) Presentation (5) Redis (5) WordPress (5) Clustering (4) Nginx (4) S3 (4) Zookeeper (4) Chef (3) Compilation (3) Hive (3) IOPS (3) Infographic (3) Latency Based Routing (3) Mahout (3) Mobile cloud (3) NAT (3) Security (3) Session Synch (3) UltraDNS (3) snapshots (3) High IO (2) Jgroups (2) Rightscale (2) Terracotta (2) flume (2) Awards (1) BAAS (1) Cloud Stack (1) IIT-DOMS (1) RabbitMQ (1) Rackspace (1) SecureCloud (1) Social Media (1) chukwa (1) mobile cloud computing (1)

**Harish Videos**

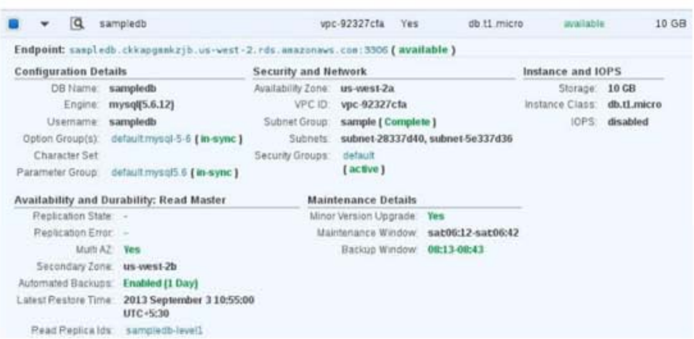
powered by YouTube

**Categories**

Compilation

Configuration

Cost Savings



sampledb vpc-92327cfa Yes db.t1.micro available 10 GB

Endpoint: sampledb.ckkapgmkzjb.us-west-2.rds.amazonaws.com:3306 ( available )

Configuration Details
DB Name: sampledb
Engine: mysql(5.6.12)
Username: sampledb
Option Group(s): default.mysql-5-6 ( in-sync )
Character Set:
Parameter Group: default.mysql5.6 ( in-sync )

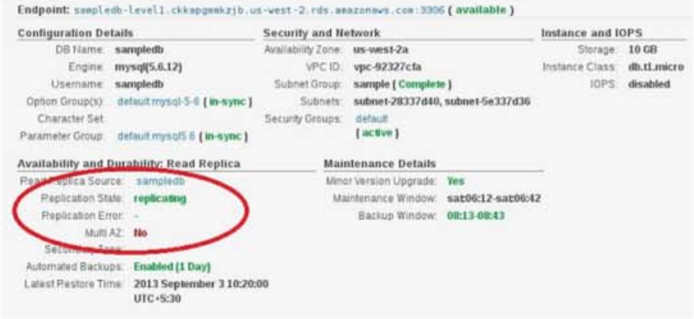
Security and Network
Availability Zone: us-west-2a
VPC ID: vpc-92327cfa
Subnet Group: sample ( Complete )
Subnets: subnet-28337d40, subnet-5e337d36
Security Groups: default ( active )

Instance and IOPS
Storage: 10 GB
Instance Class: db.t1.micro
IOPS: disabled

Availability and Durability: Read Master
Replication State: -
Replication Error: -
Multi AZ: Yes
Secondary Zone: us-west-2b
Automated Backups: Enabled (1 Day)
Latest Restore Time: 2013 September 3 10:55:00 UTC+5:30
Read Replica Ids: sampledb-level1

Maintenance Details
Minor Version Upgrade: Yes
Maintenance Window: sat06:12-sat06:42
Backup Window: 08:13-08:43

**Explore the status from Tier 1 Read Replica :** When you explore the Tier 1 Read Replica details, you will find it is pointing to the to Master DB. Illustrated in Below Screen Shot :



Endpoint: sampledb-level1.ckkapgmkzjb.us-west-2.rds.amazonaws.com:3306 ( available )

Configuration Details
DB Name: sampledb
Engine: mysql(5.6.12)
Username: sampledb
Option Group(s): default.mysql-5-6 ( in-sync )
Character Set:
Parameter Group: default.mysql5.6 ( in-sync )

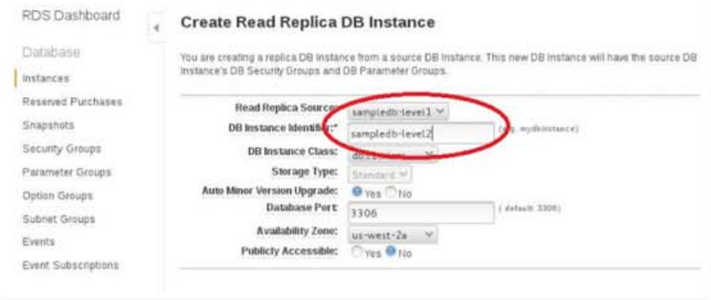
Security and Network
Availability Zone: us-west-2a
VPC ID: vpc-92327cfa
Subnet Group: sample ( Complete )
Subnets: subnet-28337d40, subnet-5e337d36
Security Groups: default ( active )

Instance and IOPS
Storage: 10 GB
Instance Class: db.t1.micro
IOPS: disabled

Availability and Durability: Read Replica
Read Replica Source: sampledb
Replication State: replicating
Replication Error: -
Multi AZ: No
Secondary Zone:
Automated Backups: Enabled (1 Day)
Latest Restore Time: 2013 September 3 10:20:00 UTC+5:30

Maintenance Details
Minor Version Upgrade: Yes
Maintenance Window: sat06:12-sat06:42
Backup Window: 08:13-08:43

**Step 2: Creating Second Tier Read Replica from Tier 1 Read replica :**
To create Second Tier Read replica navigate to the dashboard of Amazon RDS, select the Amazon RDS Tier 1 Read Replica (**sampledb-leve1**) as the **source** and use the option of "Create Read Replica".



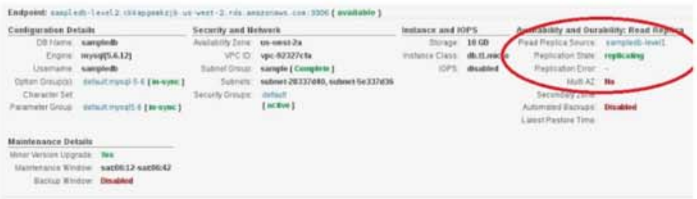
Create Read Replica DB Instance

You are creating a replica DB instance from a source DB instance. This new DB instance will have the source DB instance's DB Security Groups and DB Parameter Groups.

Read Replica Source: sampledb-level1
DB Instance Identifier:* sampledb-level2 (e.g. mydbinstance)
DB Instance Class: db.t1.micro
Storage Type: Standard
Auto Minor Version Upgrade: ● Yes ○ No
Database Port: 3306 ( default: 3306 )
Availability Zone: us-west-2a
Publicly Accessible: ○ Yes ● No

Name the newly created Amazon RDS Read replica as "**sampledb-level2**" and place it in Second Tier 2. The Second Tier -2 Amazon RDS MySQL read Replica can be created in same AZ of Tier 1/Master or in a different AZ for High Availability.

**Explore the status from Tier 2 Read Replica :** When you explore the Tier 2 Read Replica details, you will find it is pointing to the to Tier 1 - sampledb-level1 as replication source. Illustrated in Below Screen Shot :



Endpoint: sampledb-level2.ckkapgmkzjb.us-west-2.rds.amazonaws.com:3306 ( available )

Configuration Details
DB Name: sampledb
Engine: mysql(5.6.12)
Username: sampledb
Option Group(s): default.mysql-5-6 ( in-sync )
Character Set:
Parameter Group: default.mysql5.6 ( in-sync )

Security and Network
Availability Zone: us-west-2a
VPC ID: vpc-92327cfa
Subnet Group: sample ( Complete )
Subnets: subnet-28337d40, subnet-5e337d36
Security Groups: default ( active )

Instance and IOPS
Storage: 10 GB
Instance Class: db.t1.micro
IOPS: disabled

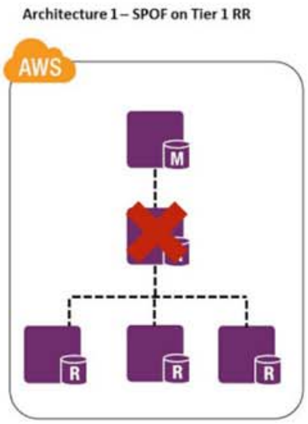
Availability and Durability: Read Replica
Read Replica Source: sampledb-level1
Replication State: replicating
Replication Error: -
Multi AZ: No
Secondary Zone:
Automated Backups: Disabled
Latest Restore Time:

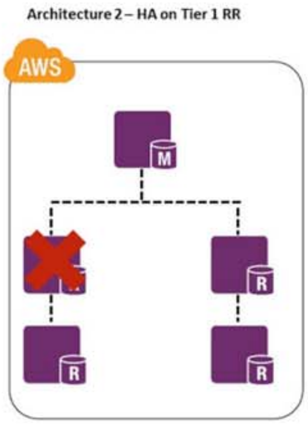
Maintenance Details
Minor Version Upgrade: Yes
Maintenance Window: sat06:12-sat06:42
Backup Window: Disabled

To Load Balance AWS RDS Read Replica's Refer this article : **http://harish11g.blogspot.com/2013/08/Load-balancing-Amazon-RDS-MySQL-read-replica-slaves-using-HAProxy.html**

## Second Tier Read Replica Deployment Architectures

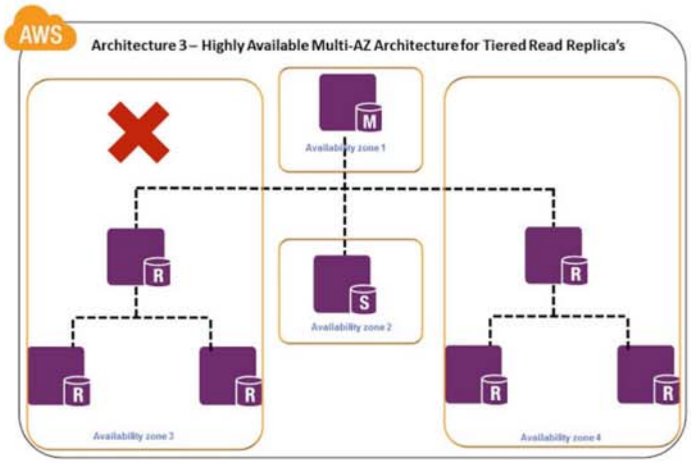
One of the main complexity behind Multi-Level replication is that, if Tier 1 Read Replica if not properly architected/placed, it can be a single point of failure. Imagine a case where you need 4 Read Replicas for your Master DB. You can take following approaches as illustrated below while designing your infrastructure for this requirement.





In Architecture-1, Tier-1 Read replica is a Single Point of Failure. Instead if you split Tier-1 itself into two separate fleets it offers better availability than architecture-1. Since both Tier-1 RR put replication load on Master DB , you can by pass this using Hot Standby instance. Benefits of this approach is explained below in the best practice architecture.

**Best Practice High Availability Architecture for Second Tier Read Replica:**



**Condition 1 - Master DB Failure :** Hot Standby becomes new master and Tier-1 Read Replicas points automatically to new source
**Condition 2- Tier-1 Read Replica Failure :** Entire Tier-1 and Associated Tier-2 has to be recreated. The Alternate Active Fleet of Tier-1+Tier-2 will serve the requests for high availability.
**Condition 3 - Tier-2 Read Replica Failure :** Only the non performing Tier-2 Read Replica instance has to be recreated
**Condition 4- AZ NW problem:** In event AZ-3 is failed, Requests are served by alternate fleet of Tier-1+Tier-2 in another AZ(AZ-4).

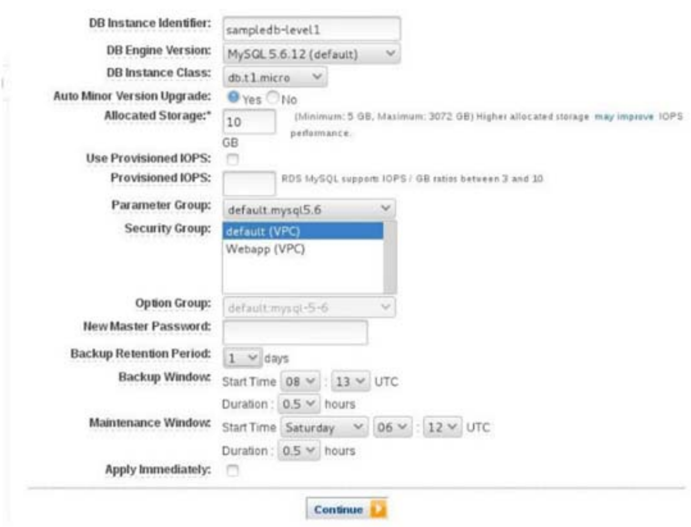
Multi-AZ Hot Standby Implementation is a recommended best practice when it comes to Multi-tiered Read replica implementation

- It is recommended to create Tier 1 Read Replicas from Multi-AZ DB instance to offload read queries from the source master DB instance for high traffic sites. If the source instance of a Multi-AZ deployment fails over to the secondary, any associated read replicas will be switched to use the secondary as their replication source automatically. This model guarantees high availability.
- Also when you initiate the creation of a Tier 1 read replica, Amazon RDS takes the DB Snapshot of your Standby DB instance (instead of Source DB) and begins replication. This model saves I/O suspension on your source DB during the snapshot process

## Other Points to Note:

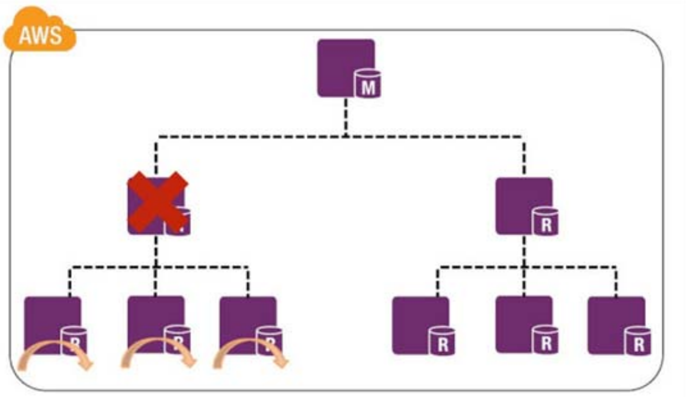
**Note :**
To Load Balance AWS RDS Read Replica's Refer this article : **http://harish11g.blogspot.com/2013/08/Load-balancing-Amazon-RDS-MySQL-read-replica-slaves-using-HAProxy.html**

**P1)** Circular replication are not allowed in this tiered replica creation process.
**P2)**Third Tier cannot be created. Actually in production, very rarely we need third tier and it is not important feature
**P3)** When the "X" Tier replica is placed in Different AZ , you should factor few extra milliseconds of latency during replication.
**P4)** Before a DB instance (Master or Tier 1) can serve as a replication source, you must enable automatic backups on the source DB instance by setting the backup retention period to a value other than 0. This requirement does not apply to second tier replica as they are not source DB instance for another read replica.So create "Read Replica" option will appear only when the Backup Retention is set to minimum of 1 day for any level Read Replica creation. When we "create Read Replica" by default RR is created with Backup Retention set to 0. Use the modify option Illustrated in below screenshot and set the Backup Retention Period. :



DB Instance Identifier: sampledb-level1
DB Engine Version: MySQL 5.6.12 (default)
DB Instance Class: db.t1.micro
Auto Minor Version Upgrade: ● Yes ○ No
Allocated Storage:* 10 GB (Minimum: 5 GB, Maximum: 3072 GB) Higher allocated storage may improve IOPS performance.
Use Provisioned IOPS: □
Provisioned IOPS: RDS MySQL supports IOPS / GB ratios between 3 and 10
Parameter Group: default.mysql5.6
Security Group: default (VPC)
Webapp (VPC)
Option Group: default:mysql-5-6
New Master Password:
Backup Retention Period: 1 days
Backup Window: Start Time 08 : 13 UTC
Duration : 0.5 hours
Maintenance Window: Start Time Saturday 06 : 12 UTC
Duration : 0.5 hours
Apply Immediately: □

**P5)** We have used t1.micro as RDS DB for article explanatory purpose. For production use cases please use proper instance types after proper capacity planning.
**P6)** AWS has released Parallel replica creation process, where you can create multiple Tier 1 and Tier 2 replicas in parallel. Since you no longer need to wait for one replica creation before starting the next one, it becomes easy to create multiple RR quickly. Without this feature, it would take hours to create a large Multi-tiered Replica setup.
**P7)** If a replica lags too far behind for your environment, the normal practice is to scale up or consider deleting and recreating the read replica. Imagine you have architecture where there are two Tier 1 and three Second Tier RR as illustrated in below architecture:



Now since Tier -1 RR is lagging , you are planning to delete and recreate the same. When you delete the Tier-1 RR, All the Second Tier Read Replica's will now become Standalone , Single AZ DB's. Once you have recreated the Tier -1 RR again and managed to retain the original end point as well, you cannot re-point these Second Tier RR to Tier -1 again. You need to recreate all 3 second tier from the Tier -1 again. For a application with heavy DB dependency and read traffic, it means 1/2 of the fleet is now down. This can lead to uncomfortable performance situation. This is not a ideal condition and AWS RDS team can take this in their Road Map. For such cases, it is better to create a new Tier -1 and Second tier fleet first , update your LB/App configs and then delete the old lagging fleet
**P8)Why it is better to have Multi-tiered Read Replica as the last resort in your architecture ?**
**8.a)**Currently 30 Read Replica can be created overall for a master in two tiers. 30 Read replica is more than sufficient and usually turns out to be costly architecture approach. Use this approach only for cases, which demand heavy read and when application code cannot accommodate changes and are highly DB dependent.
**8.b)**It is recommended in DB world to stay away from Multi-level replication as much possible. Your architecture will be much simpler with one master and "X" replica slaves, rather than having tiered replica's. As we observed in above deployment architectures, the second tier replica slave will be a trouble to manage in event of replication delay, crashes and network problems affecting the Tier-1 RR or the Master DB.
**8.c)**In case your application code can be redesigned, it is recommended to take following approaches before resorting to Second Tier replicas architecture

- Functional partition the RDS MySQL with Hot Stand By and Read Replica's
- Re-balance the DB load by using alternate data stores provided by AWS like DynamoDB, ElastiCache, CloudSearch etc.

If the above methods does not work for you, take the Second tier Read replica approach.

Thanks Senthil and Ram of 8KMiles for taking part in this analysis.

Posted by Harish Ganesan

+4 Recommend this on Google

Labels: Amazon web services, Architecture, AWS, Cloud computing, Configuration, High Availability, Multi AZ, MySQL, RDS

## No comments:

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

**Need Consulting help ?**

Name

Email *

Message *

Send

**DISCLAIMER**
**All posts, comments, views expressed in this blog are my own and does not represent the positions or views of my past, present or future employers. The intention of this blog is to share my experience and views. Content is subject to change without any notice. While I would do my best to quote the original author or copyright owners wherever I reference them, if you find any of the content / images violating copyright, please let me know and I will act upon it immediately. Lastly, I encourage you to share the content of this blog in general with other online communities for non-commercial and educational purposes.**

**Followers**

**Join this site**
with Google Friend Connect

**Members (83)**

Already a member? Sign in

**My Presentations / Webinars / Conferences**

Speaker at Cloud Connect 2013: Lock,Stock and X Smoking EC2's
Speaker at INTEROP 2012: Overcoming outages in public clouds- High Availability patterns
Speaker at Cloud Connect 2012: How enterprises are leveraging mobile + cloud computing ?
Speaker at Cloud Developer Conference 2012 :The Art of Infrastructure Elasticity
Speaker at RACSS 2012 Conference: Overview of Amazon Web Services
Presenter: Architecting your Mobile Application for the Cloud
Presenter: Prepare your IT infrastructure for Thanksgiving in AWS
Speaker at AWS Cloud Tour 2011 : Scale new business peaks with Amazon AutoScaling
Speaker at Cloud Developer Conference 2011 :High Availability architectures in AWS
NASSCOM Emergeout 2010 panelist : Selling strategies in the new Economy
Speaker at NASSCOM Emerge 2009 : Virtual Sourcing leveraging Cloud
Speaker at AWS 201: Breakout Track speaker Singapore 2012
Speaker at Cloud Computing Summit ,India 2012 : The Business Value of Mobile Cloud Computing
Guest Speaker at IIT-DOMS on Cloud Computing
Speaker at CSI IT2020:Big Data track
Workshop on Amazon Web Services @ VIT, Vellore

**Popular Posts - All Time**

Dissecting Amazon ELB : 18 things you should know

25 Best Practice Tips for architecting your Amazon VPC

Overcoming Outages in AWS : High Availability Architectures

Caching architectures using Memcached & Amazon ElastiCache

Comparison Analysis:Amazon ELB vs HAProxy EC2

High Availability @ Load Balancing Layer-HAProxy / ELB

Architecting an Highly Available and Scalable WordPress Site in AWS

Load Balancing Amazon RDS Read Replica's using HAProxy

Configuring Apache Mahout and a Clustering Example

Part 1: Understanding Amazon Elastic Block Store

**My Articles**

HighScalability.com - 7 tips for saving money on Amazon
Article on InformationWeek
DataQuest India - From being buzzwords to reality
InformationWeek India Print Issue March 2011
CloudStory.in Introduction to BigData articles
1 2 3
Cloudstory.in "Hive for Retail Analysis"
Cloudstory.in "Route53 and Geo Distributed Architectures"
HighScalability.com "The art of Infrastructure Elasticity"
Sitepoint.com "Amazon ElastiCache on Steroids"
Newvem.com "Dissecting Amazon ELB : 18 Facts you should know"
Article on Sify
Article on Rediff

**SlideShares**