

# AlwaysOn Availability Groups Failover Best Practices

**Choirul Amri**

Choirul @ gmail.com

# About Me

- Microsoft MVP SQL (2005 – 2011)
- Microsoft: Senior Consultant (2011 – 2014)



Founder: SQL Server User Group Indonesia

- <https://groups.yahoo.com/neo/groups/sqlserver-indo/info>

Co Founder: IlmuKomputer.Com

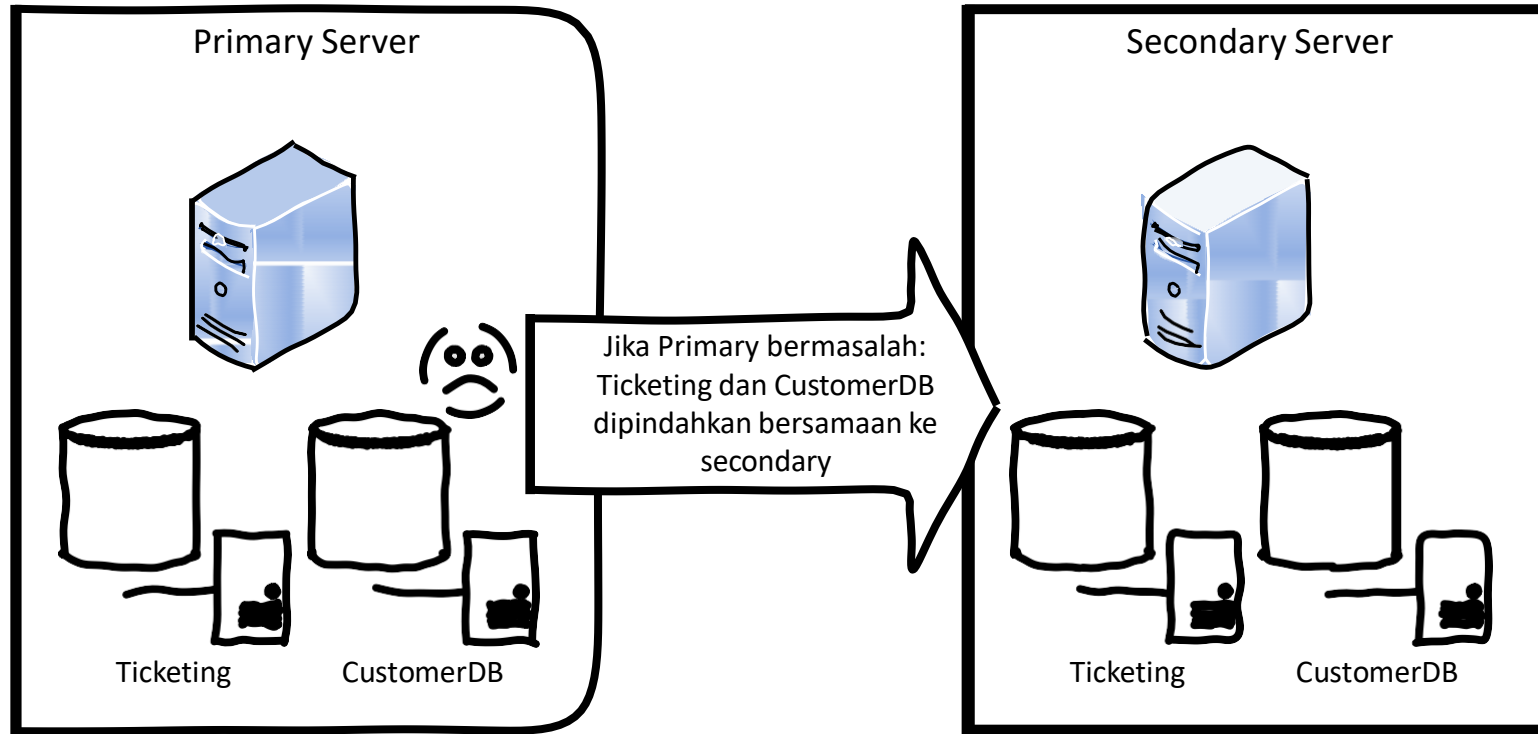
Ebiz Cipta Solusi: VP – Enterprise Division

# Session Objectives

- Key Takeaway
  - Understanding SQL Server Cluster Vs Availability Groups
  - **Design decision – best practices**
  - **Failover process and demo**
  - Focus on failover, not configuration
- What's not in session
  - Failover clustering details (unless you ask 😊)
  - Active Secondary and routing
  - Basic availability groups



# What's the "Groups" mean?



- Failover happens on "**group of databases**" rather than individual DBs
- You can also create a group with only one database
- **Why:** You want to make sure that related DBs are failed-over together

# AlwaysOn Landscape

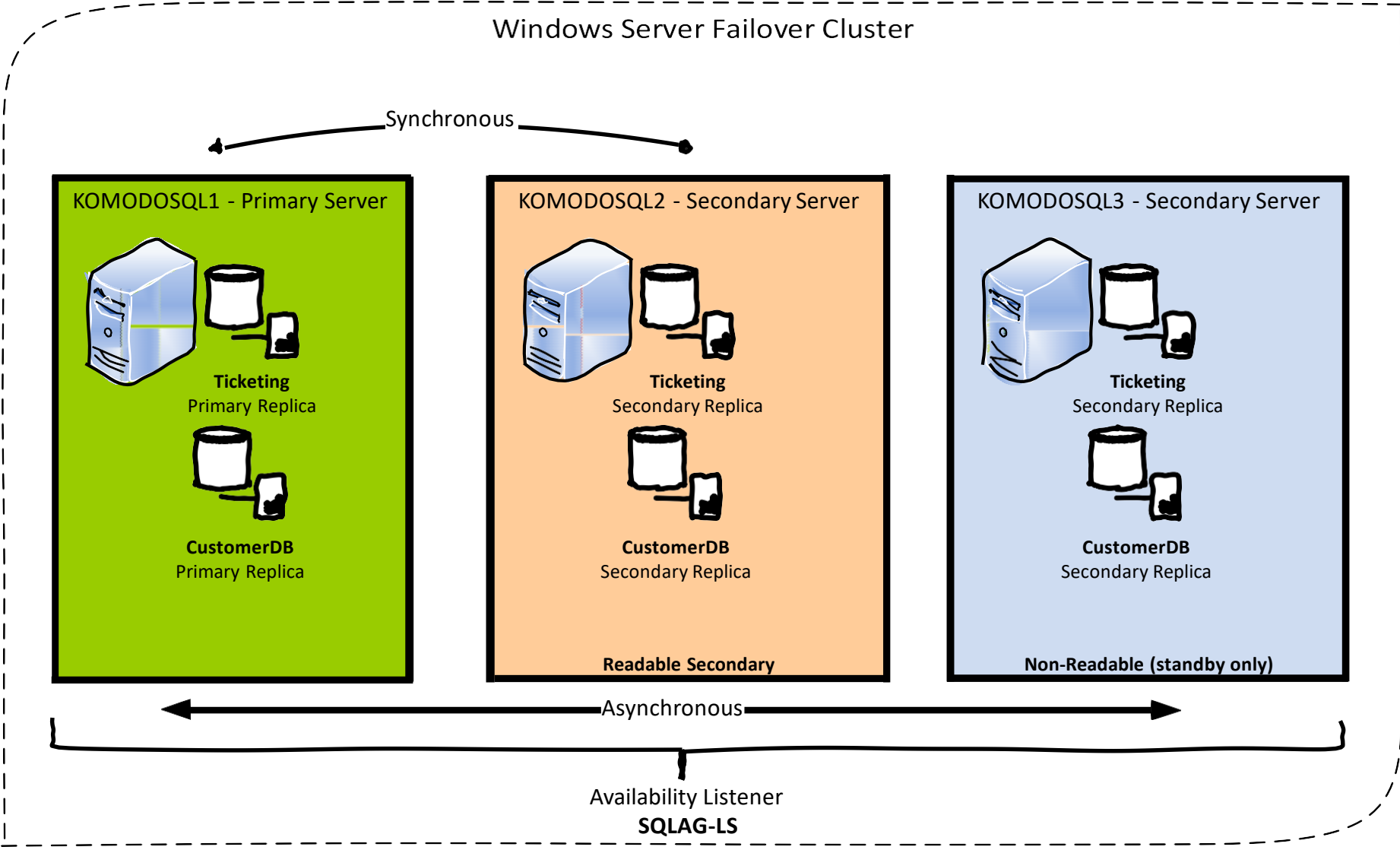
## Failover Cluster Instance (FCI)

- Instance level
- **Shared** storage
- **Failover per instance**
- No active secondary
- Support auto, transparent failover

## Availability Groups (AG)

- Database level
- **No shared** storage
- **Failover per DB group**
- Active secondary (read only)
- Support auto, transparent failover

# Availability Groups – Common Implementation



# Availability Groups Advantage

- No shared storage, easier and faster to implement
- Active secondary for better box utilization
  - Offload reporting to secondary
  - Backup from secondary
- Support automatic failover
- Transparent client failover

# Design Decision: FCI Vs AG

## Failover Cluster Instance (FCI)

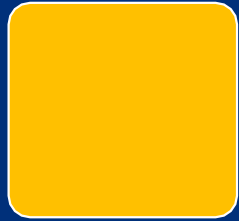
- Supporting "Legacy" apps
- Upgrading from older cluster
- Limited DBA resources
- Shared storage is available

## Availability Groups (AG)

- Apps with AG support
- Limited shared storage availability
- Isolate failover per DB
- Separate reporting is required



# Availability Groups – Availability Mode



## Asynchronous

- Better performance
- Primary commits without waiting notification from secondary
- Possible data loss in failover



## Synchronous

- Primary waits for notification from secondary before committing transaction
- Performance penalty

# Design Decision: Synchronous Vs Asynchronous

## Synchronous

- **“Good” bandwidth** is available. Start with latency 5-10ms
- Machines in the same DC
- **Automatic failover** is “absolutely” required
- Necessity instead of desire

## Asynchronous

- **DR scenario:** manual failover is a “must”
- If **network/bandwidth not “enough”**
- You can “afford” to have data loss

# Failover Mode

## Automatic



- Require **Synchronous** mode
- No data loss
- Machine shutdown or service unavailable

## Planned - Manual



- Require **Synchronous** mode
- No data loss
- Use SSMS or T-SQL

## Forced - Manual



- Replicas are in **asynchronous** commit mode
- Forced means: You can afford **losing** data
- Obviously, data loss is possible

Demo

Planned – Manual Failover

# Automatic Failover - Requirement

## Requirement

- **Minimum 1 pair** databases are in **Synchronous-commit** with Automatic failover
- Secondary databases are in **synchronized** state
- Windows Cluster is in good condition and fulfill the **quorum** requirement

Availability Replicas						
Server Instance	Role	Availability Mode	Failover Mode	Connections in Primary Role	Readable Secondary	
KOMODOSQL1\SQL2014	Primary	Synchronous commit	Automatic	Allow all conn...	Yes	
KOMODOSQL2\SQL2014	Secondary	Synchronous commit	Automatic	Allow all conn...	Read-intent only	
KOMODOSQL3\SQL2014	Secondary	Asynchronous com...	Manual	Allow all conn...	Read-intent only	

# Automatic Failover – Ready to failover



KomodoSQLAG: hosted by KOMODOSQL1\SQL2014 (Replica role: Primary)

Availability group state: Healthy  
Primary instance: KOMODOSQL1\SQL2014  
Failover mode: Automatic  
Cluster state: WIN2012-SQLAG (Normal Quorum)

Availability replica:

Name	Role	Failover Mode	Synchronization State	Issues
<a href="#">KOMODOSQL1\SQL2014</a>	Primary	Automatic	Synchronized	
<a href="#">KOMODOSQL2\SQL2014</a>	Secondary	Automatic	Synchronized	
<a href="#">KOMODOSQL3\SQL2014</a>	Secondary	Manual	Synchronized	

Group by ▾

Name	Replica	Synchronization State	Failover Readiness
<a href="#">KOMODOSQL1\SQL2014</a>			
AdventureWorks2014	KOMODOSQL1\SQL2014	Synchronized	No Data Loss
<a href="#">KOMODOSQL2\SQL2014</a>			
AdventureWorks2014	KOMODOSQL2\SQL2014	Synchronized	No Data Loss
<a href="#">KOMODOSQL3\SQL2014</a>			
AdventureWorks2014	KOMODOSQL3\SQL2014	Synchronized	No Data Loss

# Demo

# Automatic Failover

# Real life decision: Which failover to use



## Automatic

- Servers in the same data center with low latency network
- Unexpected machine crash or SQL service disruption
- Happens in the background without notice



## Planned - Manual

- Patching OS or SQL Server
- HA/DR exercise
- Upgrading hardware



## Forced - Manual

- Real disaster happens – failover to DR site
- 3 servers, but 2 machines in primary DC are down
- There is no primary available in the configuration!



# Forced Manual Failover – Fixing “Resolving” condition

Server Instance	Availability Mode	Failover Mode	Role
KOMODOSQL1\SQL2014	Synchronous commit	Automatic	Primary
KOMODOSQL2\SQL2014	Synchronous commit	Automatic	Secondary
KOMODOSQL3\SQL2014	Asynchronous commit	Manual	Secondary

What happen if:

- 1) KOMODOSQL2\SQL2014 is down; then
- 2) Client still can access the primary as usual (but now there is no “automatic” pair)
- 3) KOMODOSQL1\SQL2014 is down
- 4) Automatic failover to KOMODOSQL3\SQL2014 will not happen!
- 5) DBs in KOMODOSQL3\SQL2014 replica will be in “resolving” mode
- 6) **Client application cannot connect**

# Recovering the “last man standing” node

Connect

KOMODOSQL1\SQL2014 (SQL Server 12.0.2000)  
KOMODOSQL2\SQL2014 (SQL Server 12.0.2000)  
KOMODOSQL3\SQL2014 (SQL Server 12.0.2000)

- Databases
- Security
- Server Objects
- Replication
- AlwaysOn High Availability
  - Availability Groups
    - KomodoSQLAG (Resolving)
      - Availability Replicas
        - Availability Databases
          - AdventureWorks2014
          - SalesDB
  - Availability Group Listeners
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)

KomodoSQLAG: hosted by KOMODOSQL3\SQL2014 (Replica role: Resolving)

Availability group state: **Critical** --- Critical (1)

Primary instance: Unknown

Failover mode: Unknown

Cluster state: WIN2012-SQLAG (Normal Quorum)

Availability replica:

Name	Role	Failover Mode	Synchronization State	Availability Mode
KOMODOSQL3\SQL...	Resolving	Manual	Not Synchronizing	Asynchronous commit

Group by

Name	Replica	Synchronization State	Failover Readin...
KOMODOSQL3\SQL2014			
AdventureWorks2014	KOMODOSQL3\SQL2014	Not Synchronizing	Data Loss
SalesDB	KOMODOSQL3\SQL2014	Not Synchronizing	Data Loss

Your situation:

- **Primary** and **secondary** in production are **lost**
- You only have 1 node in DR, but it is in resolving state, inaccessible from client

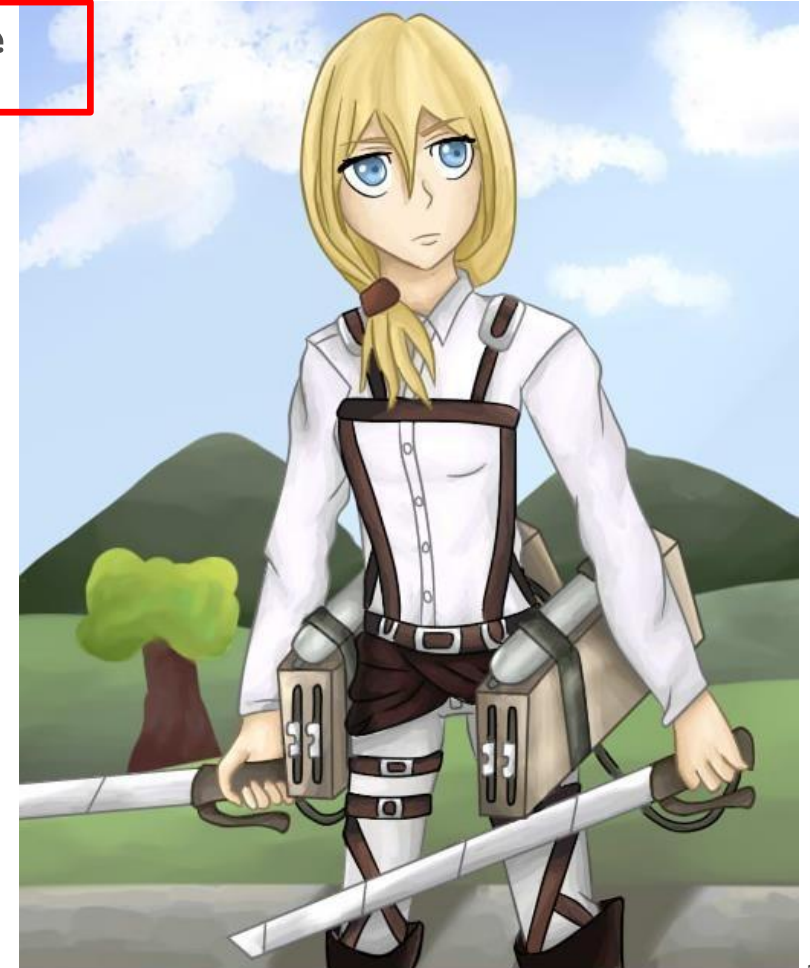
# Bring Your “resolving” replica alive

--- YOU MUST EXECUTE THE FOLLOWING SCRIPT IN SQLCMD MODE

--- KOMODOSQL3 is the replica in resolving state

:Connect KOMODOSQL3\SQL2014

```
ALTER AVAILABILITY GROUP [KomodoSQLAG]  
FORCE_FAILOVER_ALLOW_DATA_LOSS;
```



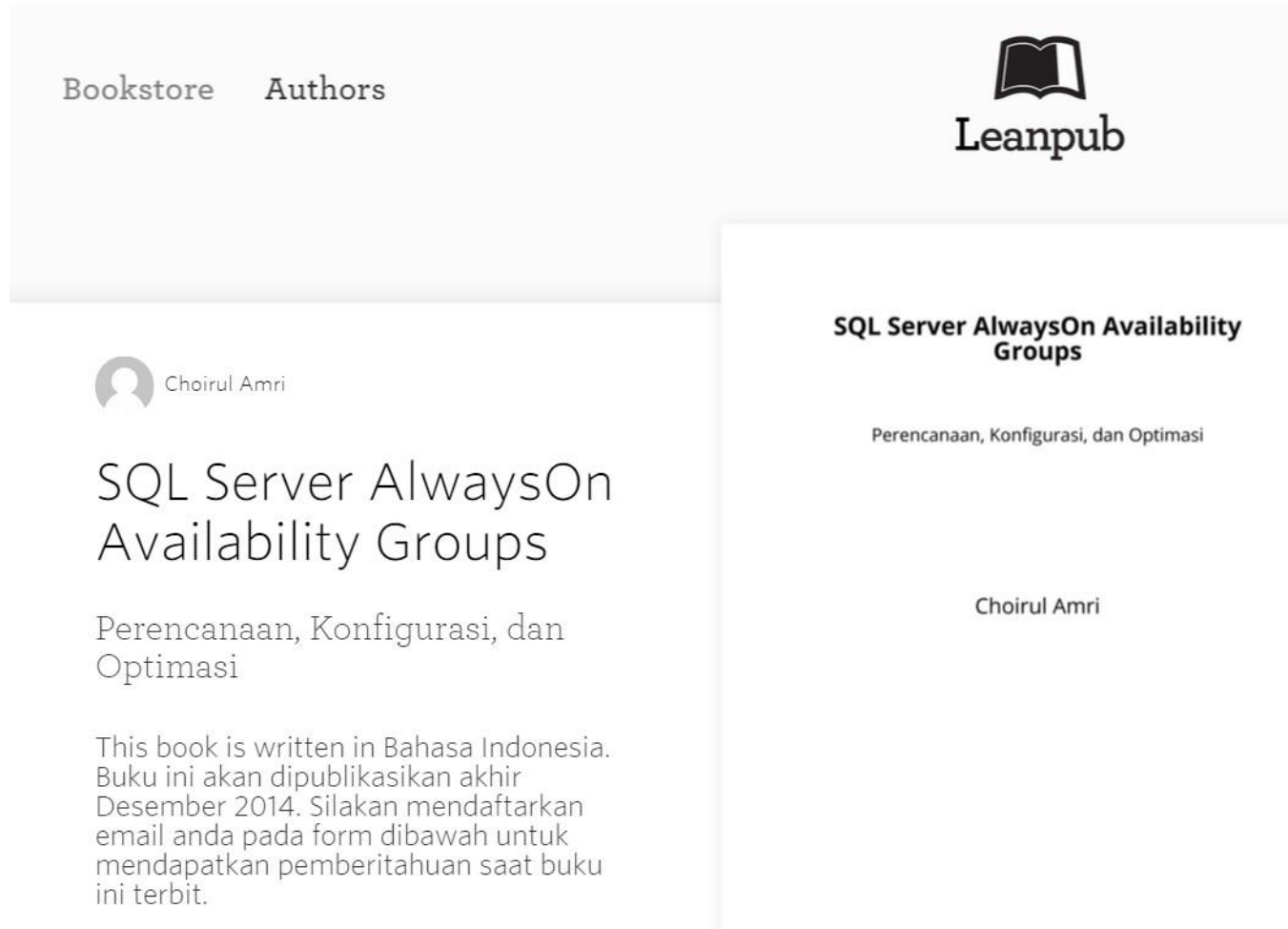
Demo

Recovering from “Resolving” state

# Recommended reference

- <http://blogs.msdn.com/b/alwaysonpro/archive/2014/08/29/recommendations-and-best-practices-when-deploying-sql-server-always-on-availability-groups-in-windows-azure-iaas.aspx>
- <http://www.brentozar.com/sql/sql-server-always-on-availability-Groups>

# Download my e-book 😊



The screenshot shows the Leanpub website interface. At the top, there are navigation links for 'Bookstore' and 'Authors', and the Leanpub logo (an open book icon) with the text 'Leanpub'. Below this, the author's profile is displayed, featuring a circular profile picture and the name 'Choirul Amri'. The main content area shows the book title 'SQL Server AlwaysOn Availability Groups' in a large font, followed by the subtitle 'Perencanaan, Konfigurasi, dan Optimasi' in a smaller font. Below the subtitle, the author's name 'Choirul Amri' is listed. At the bottom of the book preview, there is a paragraph of text in Indonesian: 'This book is written in Bahasa Indonesia. Buku ini akan dipublikasikan akhir Desember 2014. Silakan mendaftarkan email anda pada form dibawah untuk mendapatkan pemberitahuan saat buku ini terbit.'

<https://leanpub.com/sqlagindo>

Work in progress, If You find **typos** please @mention me at **@mchoirul**