



# dbWatch Control Center



## POC Guide

*Document Changes*

*V1.0*

*Original Document*

## Content

<b>Chapter 1 Introduction</b> .....	<b>3</b>
<b>Chapter 2 Prerequisites for POC</b> .....	<b>4</b>
<b>Chapter 3 Architecture</b> .....	<b>5</b>
<b>Agent-less monitoring and management</b> .....	<b>6</b>
<b>Distributed repository</b> .....	<b>6</b>
<b>Distributed and open platform</b> .....	<b>6</b>
<b>dbWatch Engines</b> .....	<b>7</b>
<b>dbWatch Control Center Server</b> .....	<b>8</b>
<b>UX Client</b> .....	<b>9</b>
<b>Chapter 4 Distributed monitoring</b> .....	<b>10</b>

# CHAPTER 1 INTRODUCTION

This document is designed to work together with the installation guide, to help guide a customer in the process of a Prof Of Concept (POC) setup, to get started the best way possible.

dbWatch Control Center POC installations is currently only supported on the Windows platform.

The setup of a dbWatch Server is found in the platform installation guide document.

## CHAPTER 2 PREREQUISITES FOR POC

Technical requirements for the dbWatch Server is found in the installation guide document.

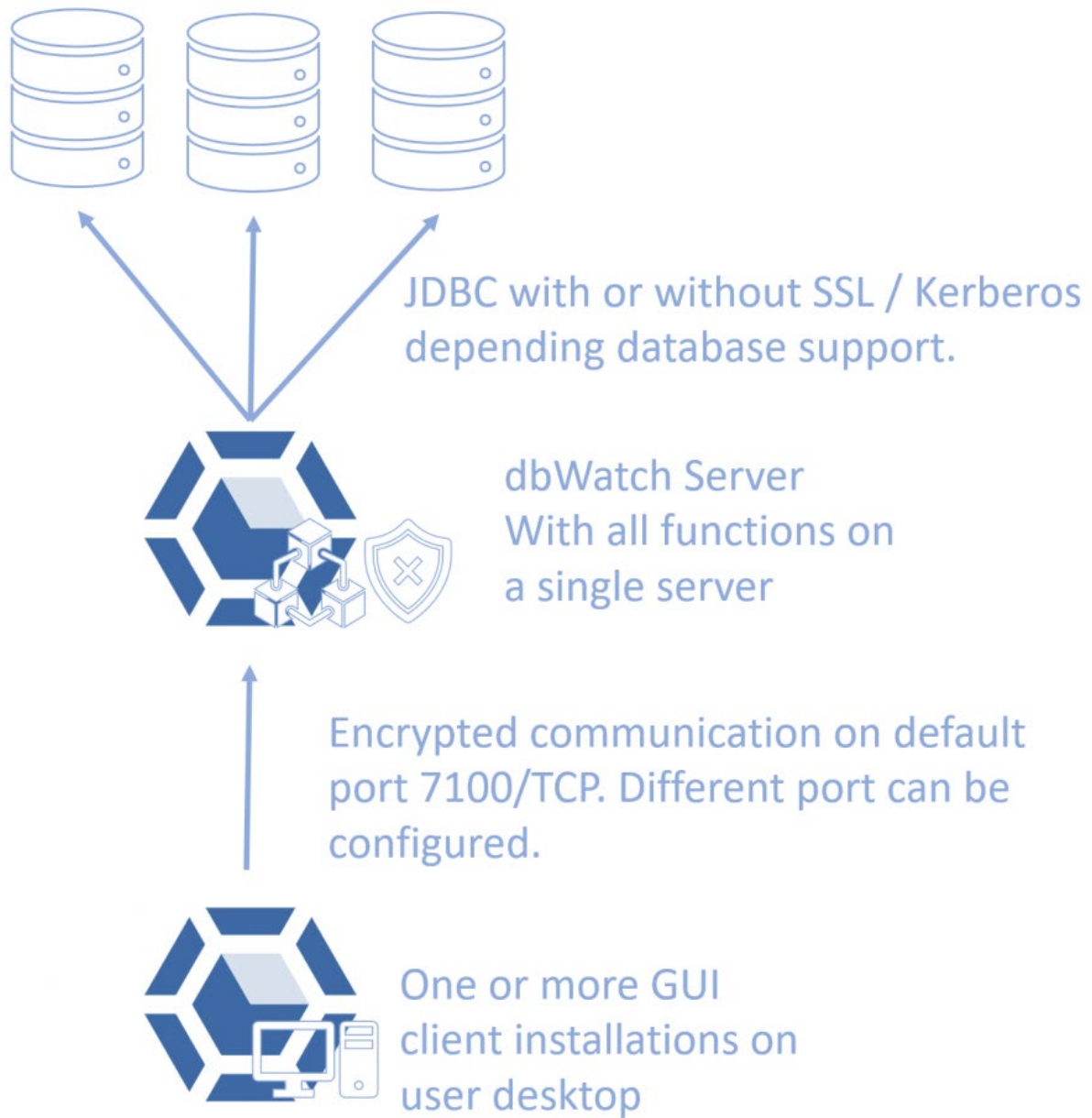
A basic setup with one dbWatch Server covering all functions and the client software either used from the dbWatch Server or from already existing clients, require only one new server. This is the setup described in the basic setup in the Architecture chapter.

Firewall must allow connections from the dbWatch Server to each of the database instances you want to monitor on their JDBC port. Typically 1433/TCP (SQL Server), 1521/TCP (Oracle), 5432/TCP (PostgreSQL), 3306/TCP (MySQL/MariaDB).

The firewall must also allow connection from the client installations to dbWatch Server on 7100/TCP.

# CHAPTER 3 ARCHITECTURE

Example drawing for a basic setup:



## Agent-less monitoring and management

**dbWatch Control Center** supports two ways of monitoring and managing database servers:

- "Basic mode." No agents, software, or files must be installed on the database server instance, underlying operating system, or server.
- "Standard mode." A small schema is installed on the database instance.

In neither mode is any agent required. Installation and operation require only access to the database server.

## Distributed repository

The statistics collected are stored and distributed on each monitored instance, and automatic maintenance procedures ensure the log files and collected data are kept relevant and compact.

## Distributed and open platform

**Control Center** is built on a unique architecture that provides the best possible capabilities for monitoring and managing heterogeneous database platforms.

Our design provides the most precise and relevant database monitoring information possible. With **Control Center**, you can modify any of the included monitoring procedures to suit your preferences better.

But we took it a step further—we designed **Control Center** so you'll have the power to design and develop your own monitoring procedures and jobs from scratch, fully customized to your needs, technical requirements, or your company's business logic.

This openness is not limited to the monitoring procedures; also, the Management module and Reporting module provide customers with the flexibility to develop and customize how they manage or report on their database systems.

**dbWatch Control Center** is built on standard, three-tier client-server architecture, with database monitoring agents inside the database (dbWatch Engines), the application server (**dbWatch Control Center** Server), and the client GUI (**dbWatch Control Center** Monitor Client). In addition, **Control Center** includes extensions to support integration with third-party systems management solutions.

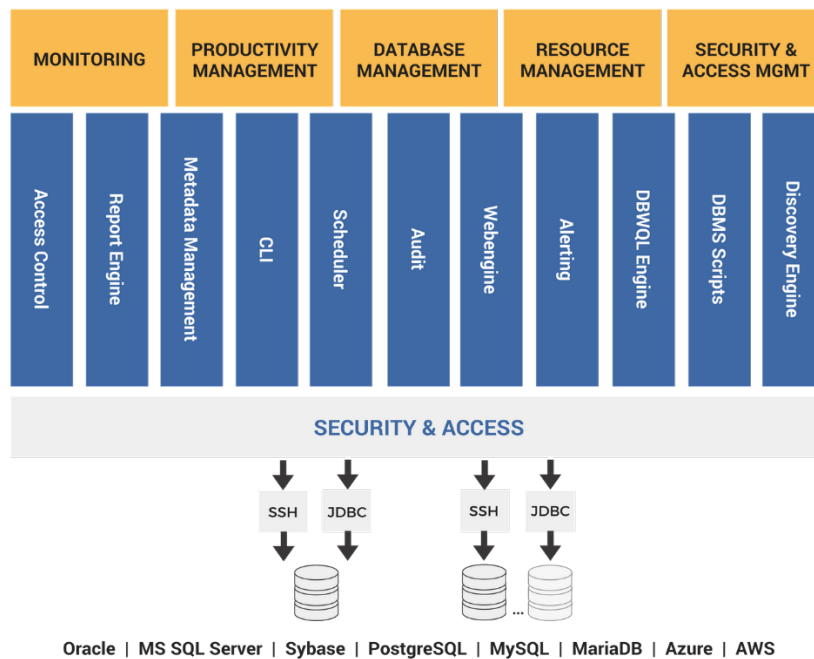


Figure 1: dbWatch Enterprise Manager overview

## dbWatch Engines

**dbWatch Control Center** uses an **intelligent database schema** inside the database, called *dbWatch Engines*, for monitoring each database instance.

Each engine is populated with a set of monitoring jobs *defined by you* as relevant to monitoring your database instances.

**dbWatch Control Center** comes with a broad range of advanced out-of-the-box monitoring jobs for each supported database platform. You can use these in their standard configurations, edit and modify them, or create your own fully customized jobs from scratch.

**Intelligent database schema.** Instead of using external agents with proprietary code, installed at the operating system level, the dbWatch engine is an intelligent database schema installed internally in the database instance being monitored, containing programmable logic, procedures, and routines for monitoring and data collection (at around 50 MB, they are completely non-intrusive and without any significant performance impact). This allows for rapid deployment **without** the hassle of installing software on each database server.

**Native support.** The dbWatch Engines are programmed in the native programming language for each supported database (PL/SQL for Oracle, T-SQL for MS SQL and Sybase, etc.). Since this is a language you are already familiar with, you will not need new training or external consultants. This also makes the dbWatch engine truly platform-independent and much easier to manage.

**Open and extensible.** The dbWatch Engines code base is fully open. You can read, edit, or modify the code exactly the way you like. And most importantly, you can create monitoring procedures and jobs that are fully customized to your work needs or your business processes.

The combination of the following features gives you a variety of advantages that no other database monitoring solution can:

- The most advanced monitoring possible—if you cannot do it in the native procedure language, you cannot do it at all.
- Fully self-service—create and modify your monitoring activities precisely how you want them to be!

### **dbWatch Control Center Server**

The **dbWatch Control Center** Server is the nerve center of the dbWatch infrastructure.

**dbWatch Control Center** Server connects and manages all dbWatch Engines and is used for the following:

- Configuring and scheduling monitoring jobs
- Alert Messaging—SMS, email, third-party tools
- Reporting—configuring, creating, generating, and distributing reports
- User administration, role-based access controls
- Security handling – encryption, certificates
- Kerberos integration
- Auto-scanning for new instances

**Scheduling and triggering monitoring jobs.** Each dbWatch job comes with a default schedule. **dbWatch Control Center** server lets you easily configure these schedules and triggers to fit your requirements and preferences (the crontab format may be new to MS users, but explanatory text fields are included).

**Alert Messaging.** **dbWatch Control Center** server is used to specify alert messaging and routing. You can configure alarms to be sent to one or multiple destinations:

- **dbWatch Control Center** client
- Email
- SMS
- Third-party system management platform

**dbWatch Control Center** alert messaging is entirely flexible, allowing you to configure different messaging rules depending on the day of the week, hour, calendar date, etc., to support the shift schedule of your IT management processes fully.

**Platform.** While **dbWatch Control Center** comes complete with all the monitoring alerts, maintenance tasks, and reports that most users need, we recognize the need for some users to adapt and extend the tools for unique requirements. **dbWatch Control Center** is delivered with an integrated development environment (IDE) and the source to all monitoring jobs so each DBA can easily modify, extend or create new jobs,



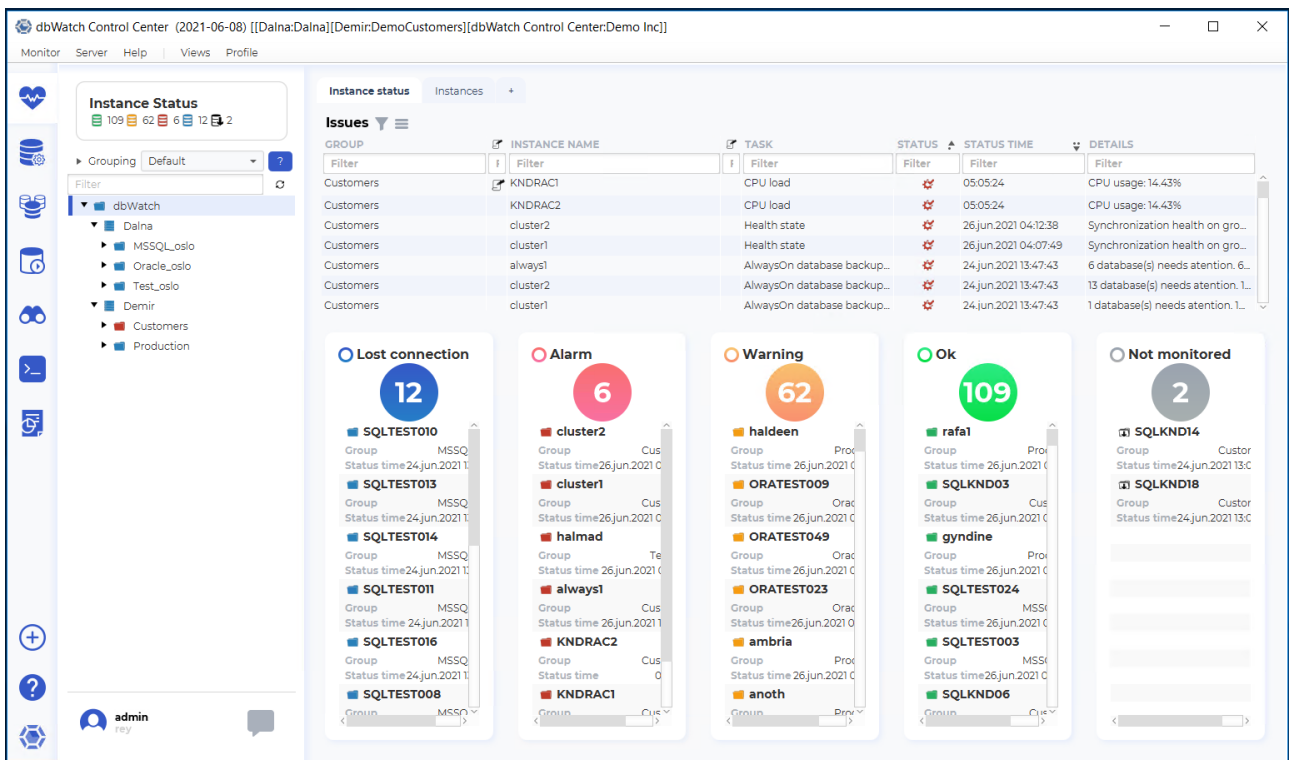
procedures, and reports. **Control Center** also includes a command line interface (CLI) and farm data language (FDL) for querying across multiple instances and versions.

### UX Client

The **dbWatch Control Center** client provides the graphical user interface for the dbWatch user with a highly intuitive and easy-to-use structure.

The client provides a complete overview of all monitored databases in a single standard interface. For ease of use, instances can be given names and properties, grouped, sorted, filtered in any way so each view matches the local needs and makes operations easier.

The complete user interface is customizable through [FDL](#). You can read about it by following the link.



# CHAPTER 4 DISTRIBUTED MONITORING

In large or complex environments, deploying multiple dbWatch servers and connecting them together in a single monitoring view is sometimes needed.

This need can arise when you:

- Need to monitor more than 250 instances (windows)
- Multiple locations/data centers
- Have subnets with servers behind firewalls
- Multiple organizations/customers/units
- Need to minimize network/Firewall traffic
- Need Redundancy

**dbWatch Control Center** has supported multiple monitoring/management servers for a long time. This can have numerous benefits:

- Scales to thousands of instances
- Simplify access to instances behind firewalls
- Reduce network traffic
- Redundancy
- Data and stats are cached locally on each server



## Single dbWatch Control Center Server

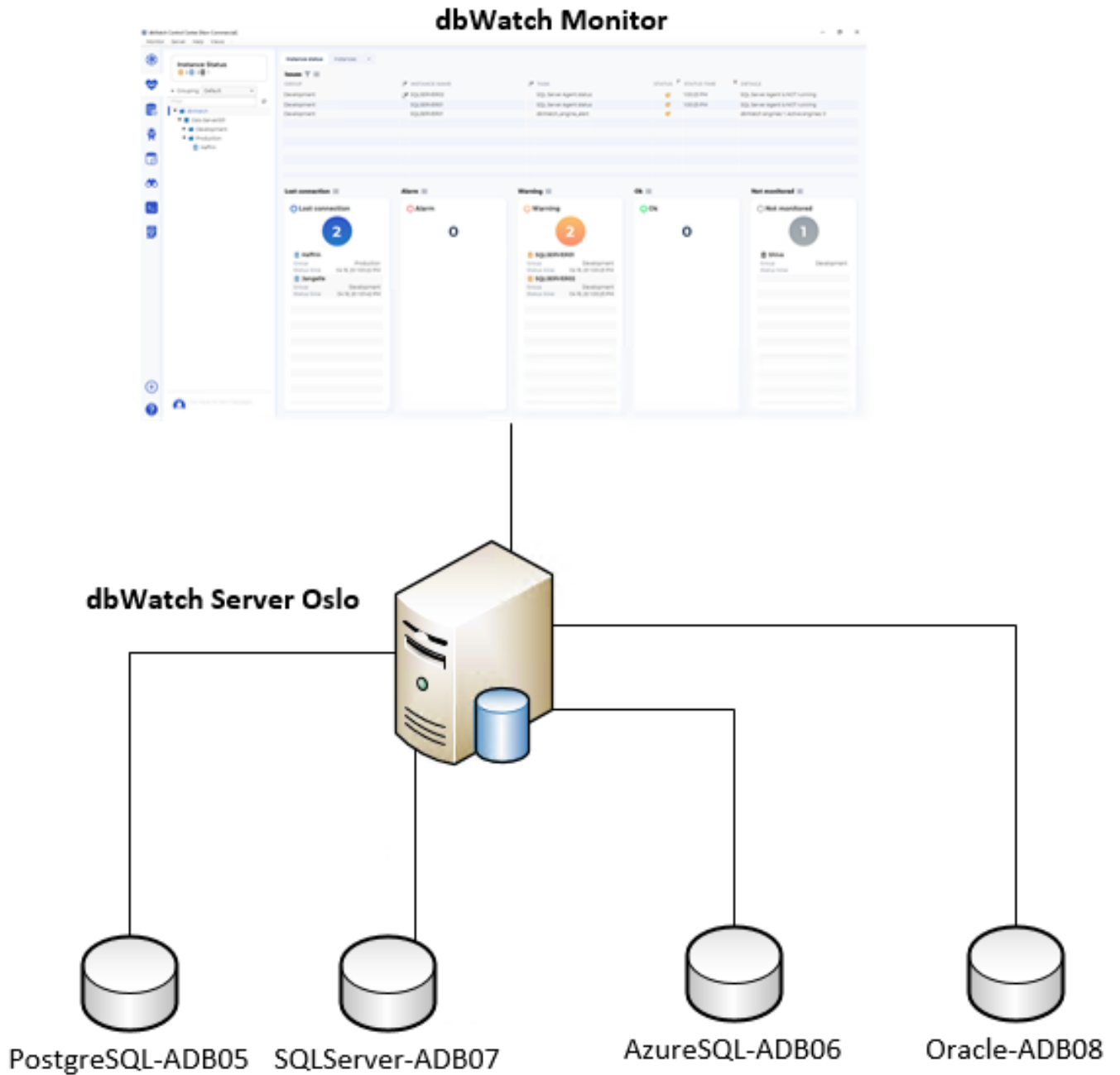


Figure 8: Normal, single server configuration. Scales to 250 (windows) instances

## Multiple dbWatch Control Center servers

### dbWatch Monitor

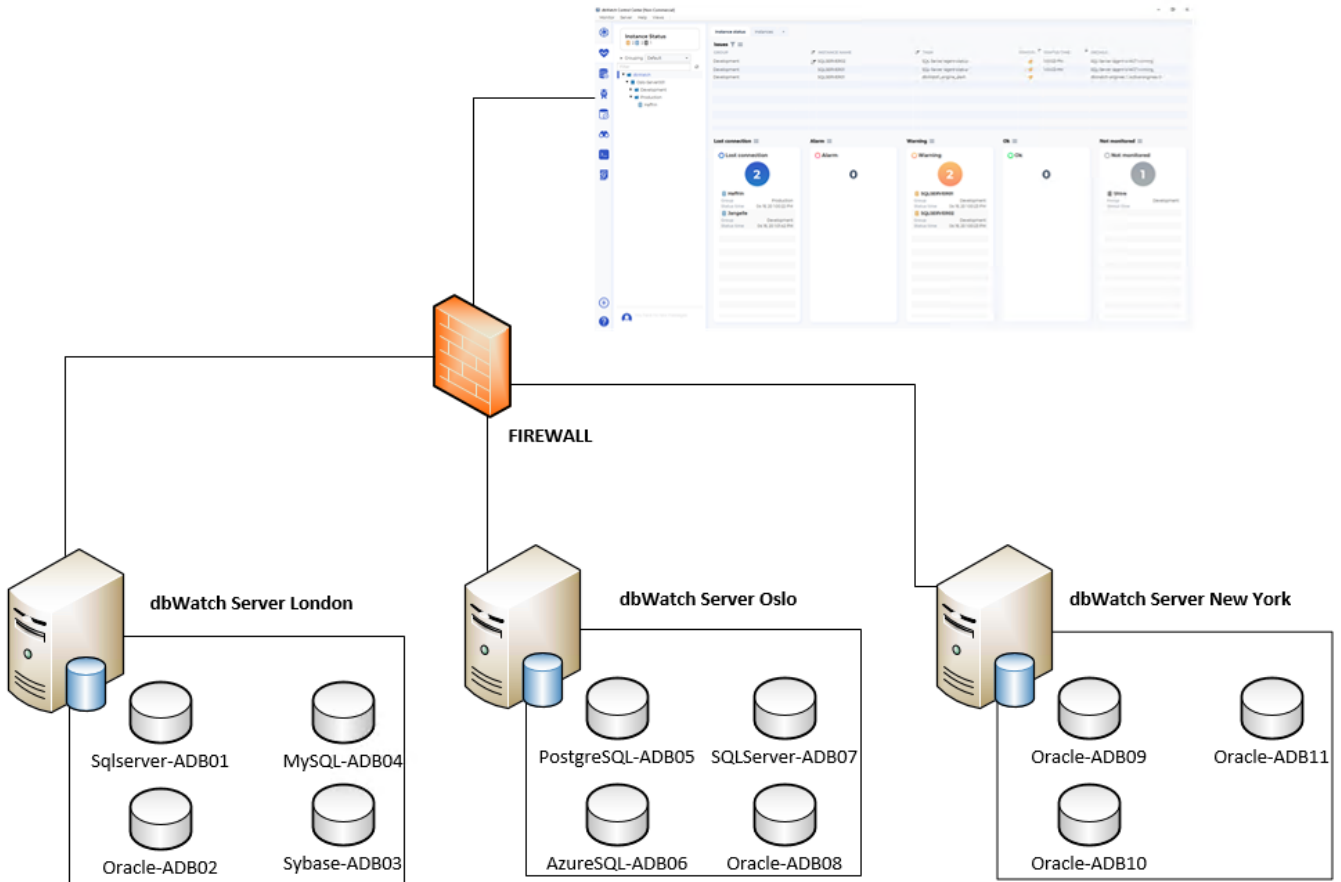


Figure 9: Multiple dbWatch servers connected in a hierarchy to serve multiple locations or subnets. Excellent scalability!