# 1.8.0

# [Infrastructure Architecture](#id-1.8.0-InfrastructureArchitecture)

# [The 3-Tier Architecture](#id-1.8.0-The3-TierArchitecture)

# [Deployable Components per Tier](#id-1.8.0-DeployableComponentsperTier)

# [Backup Procedure](#id-1.8.0-BackupProcedure)

# [Databases](#id-1.8.0-Databases)

# [BEJS\_Core](#id-1.8.0-BEJS_Core)

# [BEJS\_Utils](#id-1.8.0-BEJS_Utils)

# [Applications](#id-1.8.0-Applications)

# [Internal Portal API](#id-1.8.0-InternalPortalAPI)

# [Internal Portal UI](#id-1.8.0-InternalPortalUI)

# [External Portal API](#id-1.8.0-ExternalPortalAPI)

# [External Portal UI](#id-1.8.0-ExternalPortalUI)

# [Worker Service](#id-1.8.0-WorkerService)

# [Dhamen Service Callback API](#id-1.8.0-DhamenServiceCallbackAPI)

# [Application Event Bus Worker Service](#id-1.8.0-ApplicationEventBusWorkerServi)

# [Release](#id-1.8.0-Release)

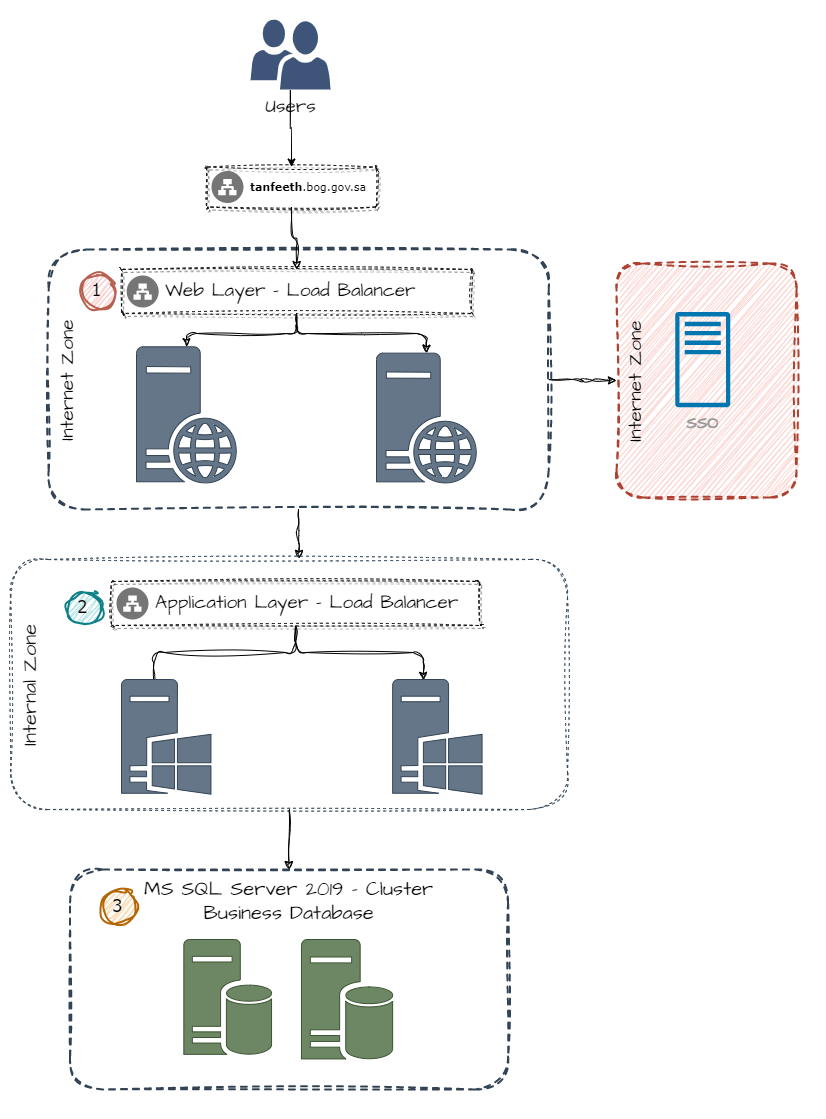
# ****Infrastructure Architecture****

## ****The 3-Tier Architecture****

The diagram below describes a high-level architecture for a staging / production infrastructure. We have adopted a **three-tier architecture** that logically and physically segregates the infrastructure into three tiers, each running on its own infrastructure. This allows each tier to be updated or scaled independently without affecting the other tiers.

The separation of concerns among the three tiers provides loose coupling, independent scalability, and improved security. The web tier can scale to handle increasing web traffic, the application tier can scale based on application load, and the data tier can scale based on data storage and I/O needs, without changes rippling through the other tiers. The isolation of the data tier also enhances security by limiting direct data access.

|  |  |  |
| --- | --- | --- |
| **#** | **Tier** | **Description** |
| 1 | **Web Tier (Web Servers)** | Contain Portals (Angular) & API Gateway |
| 2 | **Application Tier** | Contain all Web APIs that will be used for data processing. |
| 3 | **Data Tier** | contain MS Server Databases |



## ****Deployable Components per Tier****

Here a list of all deployable components that are distributed to each tier.

In the diagram above, each server is assigned a number to serve as a reference in the details.

| **Reference Id** | **Component** | **Package File Name** | **Target Tier** |
| --- | --- | --- | --- |
| **C-01** | **Database Core** | **Database.Core.[ReleaseFormat]** | **Data Tier** |
| **C-02** | **Database Utils** | **Database.Utils.[ReleaseFormat]** | **Data Tier** |
| **C-03** | **Internal Portal API** | **Internal.Portal.API.[ReleaseFormat]** | **Application Tier** |
| **C-04** | **External Portal API** | **External.Portal.API.[ReleaseFormat]** | **Application Tier** |
| **C-05** | **Worker Service** | **Worker.Service.[ReleaseFormat]** | **Application Tier** |
| **C-06** | **External Portal UI** | **External.Portal.UI.[ReleaseFormat]** | **Web Tier (Web Servers)** |
| **C-08** | **Internal Portal UI** | **Internal.Portal.UI.[ReleaseFormat]** | **Web Tier (Web Servers)** |
| **C-09** | **Application Event Bus Worker Service** | **Tanfeeth.EventBus.Application.WorkerService.[ReleaseFormat]** | **Application Tier** |
| **C-10** | **Dhamen Service Callback** | **BEJS.Services.DhamenServiceCallBack.[ReleaseFormat]** | **Application Tier** |

Note: we will refer to each component by **"Reference Id"** in all the below steps.

## ****Backup Procedure****

| **Action** | **Team** |
| --- | --- |
| Backup Databases ( **BEJS\_Core, BEJS\_Utils**) | Database |
| Backup IIS Web Site (**BEJS-Internal-UI**) | Application |
| Backup IIS Web Site (**BEJS-Internal-API**) | Application |
| Backup IIS Web Site (**BEJS-External-UI**) | Application |
| Backup IIS Web Site (**BEJS-External-API**) | Application |
| Backup IIS Web Site (**BEJS-DhamenServiceCallBack-API**) | Application |
| Backup **Application Event Bus Worker Service** | Application |
| Backup **Worker Service** | Application |

**🛠️ Installation Procedure**

## ****Databases****

| # | **Action** |
| --- | --- |
| 1 | Copy the Component Files **(C-01)** to any server that is exists in **Application Tier** |
| 2 | ****BEJS****\_Core Extract **(C-01)** release package to **physical path** then replace the "**appsettings.json**", "**NLog.config**" files with the same files from the latest backup.  example physical path: **C:\BEJS\Database.Core.MigratorUI**   * Run **DB.Core.MigratorUI.exe** * Open Browser then navigate to URL: [**http://localhost:5014**](http://localhost:5014). * Click on "**Run Migrator**" button. |
| 3 | BEJS\_Utils Extract **(C-02)** release package to **physical path** then replace the "**appsettings.json**", "**NLog.config**" files with the same files from the latest backup.  example physical path: **C:\BEJS\Database.Util.MigratorUI**   * Run **DB.Util.MigratorUI.exe** * Open Browser then navigate to URL**:** [**http://localhost:5015**](http://localhost:5015)**.** * Click on "**Run Migrator**" button. |

## ****Applications****

### ****Internal Portal API****

Please apply the below Steps for all servers exists in Application Tier

| # | **Action** | **Team** | **Comment** |
| --- | --- | --- | --- |
| 1 | Stop IIS  Run **cmd** as administrator → **iisreset /stop** | Application |  |
| 2 | Delete all files except "**appsetting.json**", "**appsetting-shared.json**", "**NLog.config**" in **C:\inetpub\wwwroot\BEJS-Internal-API** | Application |  |
| 3 | Download **Internal Portal API** release package (**C-03**) and remove "**appsetting.json**", "**appsetting-shared.json**", "**NLog.config**" then extract to physical path in server  Server Path: **C:\inetpub\wwwroot\BEJS-Internal-API** | Application |  |
|  | Open "appsetting-shared.json" file and make sure the below json object on the root with the below values   * Cap   "Cap": {      "DefaultGroupName": "tanfeeth",      "FailedRetryCount": 10,      "UseStorageLock": **true**,      "ConsumerThreadCount": 1,      "SucceedMessageExpiredAfterDays": 90,      "FailedMessageExpiredAfterDays": 180,      "FailedRetryIntervalInSeconds": 300,      "ExchangeName": "Tanfeeth.EventBus",      "UseCAPDashboard": **true**  } |  |  |
| 4 | Start IIS |  |  |

### ****Internal Portal UI****

Please do the below Steps for all servers exists in both Presentation Tier

| 1 | **Action** | **Team** | **Comment** |
| --- | --- | --- | --- |
| 2 | Download **Internal-UI** release package (**C-08**) and extract to physical path in server.  physical path: **C:\inetpub\wwwroot\BEJS-Internal-UI** | Application |  |
| 3 | Open folder "assets" located in the physical path defined in step 2 then replace the file "config.json" with the same file from the latest backup. |  |  |

### ****External Portal API****

Please apply the below Steps for all servers exists in Application Tier

| # | **Action** | **Team** | **Comment** |
| --- | --- | --- | --- |
| 1 | Stop IIS  Run **cmd** as administrator → **iisreset /stop** | Application |  |
| 2 | Delete all files except "**appsetting.json**", "**appsetting-shared.json**", "**NLog.config**" in **C:\inetpub\wwwroot\BEJS-External-API** | Application |  |
| 3 | Download **External Portal API** release package (**C-04**) and remove "**appsetting.json**", "**appsetting-shared.json**", "**NLog.config**" then extract to physical path in server  Server Path: **C:\inetpub\wwwroot\BEJS-External-API** | Application |  |
| 4 | Start IIS |  |  |

### ****External Portal UI****

Please do the below Steps for all servers exists in both Presentation Tier

| 1 | **Action** | **Team** | **Comment** |
| --- | --- | --- | --- |
| 2 | Download **External-UI** release package (**C-06**) and extract to physical path in server.  physical path: **C:\inetpub\wwwroot\BEJS-External-UI** | Application |  |
| 3 | Open folder "assets" located in the physical path defined in step 2 then replace the file "config.json" with the same file from the latest backup. |  |  |
| 4 | Save the file and close it |  |  |

### ****Worker Service****

Please apply the below Steps for all servers exists in **Application Tier**

| 1 | **Action** |
| --- | --- |
| 2 | Stop the installed Worker Service. |
| 3 | Copy release file of "Worker Service "**C-05"** and thenExtract the file to any preferred folder, suggested path " **C:\Tanfeeth\WorkerService**" |
| 4 | Delete **"NLog.config"** and "**appsetting.json**" then replace the file from the extract path to the installed service worker path. |
| 5 | Open "**appsettings.json**" file and edit   * BackgroundServices.Services   "BackgroundServices": {  "Services": [  {  "Name": "UpdateCourtMemberStatus",  "RecurringTimeInterval": "0 0 \* \* \*",  "IsActive": true  }  ]  }, |
| 6 | Start windows service for "**Tanfeeth** Windows Service" in windows service. |

### ****Dhamen Service Callback API****

| # | **Action** | **Team** | **Comment** |
| --- | --- | --- | --- |
| 1 | Stop IIS  Run **cmd** as administrator → **iisreset /stop** | Application |  |
| 2 | Delete all files except "**appsetting.json**", "**appsetting-shared.json**", "**NLog.config**" in **C:\inetpub\wwwroot\BEJS-DhamenServiceCallBack-API** | Application |  |
| 3 | Download **Dhamen Service Callback API** release package (**C-10**) and remove "**appsetting.json**", "**appsetting-shared.json**", "**NLog.config**" then extract to physical path in server  Server Path: **C:\inetpub\wwwroot\BEJS-DhamenServiceCallBack-API** | Application |  |
| 5 | Start IIS | Application |  |

### ****Application Event Bus Worker Service****

| # | **Action** | **Team** | **Comment** |
| --- | --- | --- | --- |
| 1 | Stop the installed **Application Event Bus Worker Service**. |  |  |
| 2 | Copy release file of "Worker Service "**C-09"**and thenExtract the file to any preferred folder, suggested path "**C:\BEJS\ApplicationEventBusWorkerService**" |  |  |
| 3 | Delete the file "**appsettings.json**" and Delete **"NLog.config"** and then replace the files from the extract path to the installed service worker path. |  |  |
| 4 | Start windows service for "**Application Event Bus Worker Service**" in windows service. |  |  |

## ****Release****

| **#** | **Component** |
| --- | --- |
| **1** | **Core Database** |
| **2** | **Migrator Util** |
| **3** | **External Portal API** |
| **4** | **External Portal UI** |
| **5** | **Internal Portal API** |
| **6** | **Internal Portal UI** |
| **7** | **Worker Service** |
| **8** | **Dhamen Service Callback** |
| **9** | **Application Event Bus Worker Service** |