**Bodhi Application Deployment Guide for Raspberry Pi**

**I. Install Package**

We provide two types of installation artifacts based on the framework language: .NET (dotnet) and NodeJs (nodejs).

* **NodeJs Suite**: For fast development.
* **Dotnet Suite**: For QA and Production, deployed to Kira Instruments.

The scripts package bodhi-apps-scripts.zip contains two main files and a library:

1. A PowerShell script to bootstrap the process (bodhi-apps.ps1).
2. A shell (bash) script to run on the target machine (bodhi-apps.sh).
3. A lib sub-folder containing multiple bash scripts called by bodhi-apps.sh

**Applications Included in the install .zip File**

**NodeJs:**

1. bodhi-api: A NodeJs application.
2. bodhi-ui: A static React application.
3. swp-pyreader: A Python application implementing the calculations engine.

**Dotnet:**

1. The Control Service (control)
2. Api Server (api)
3. Ui Web Server (ui)
4. Web Socket Server (ws)
5. Simple Identity Provider Server (auth)
6. bodhi-ui: A static React application.
7. swp-pyreader: A Python application implementing the calculations engine.
8. PyRunner: a python application

**II. Prerequisites**

**On the Windows PC:**

* PowerShell and PuTTY must be installed.
* PowerShell execution must be enabled.

Open a cmd window as Administrator and type ‘powershell’

OR open a PowerShell terminal as Administrator

Get-ExecutionPolicy

Set-ExecutionPolicy -ExecutionPolicy Unrestricted

**On the Raspberry Pi:**

* Must be a 64-bit Bookworm OS.
* Required dependencies:
  1. Python 3.11
  2. PostgreSQL
  3. Nginx
  4. NodeJs (only if deploying NodeJs applications)
  5. Zip and unzip

**Notes:**

* There is no need to deploy .NET 8 framework or runtime; all dotnet applications are self-contained.
* The script will set the locale to en\_US.UTF-8 and time to UTC.
* The script will manage databases creation and seeding.

**III. Installation Process**

**On the Windows PC:**

1. Download the install files from:

<https://qa-dotnet-edge.alpha.bodhi.revvitycloud.net/>

or  
<https://dev-dotnet-edge.alpha.bodhi.revvitycloud.net/>

1. Login with the access code. The web app will write a cookie valid for 30 days.
2. On the home page, press the “Bodhi Apps Install” button.
3. Click on the file names to download: the install .zip and the bodhi-apps-scripts-[version].zip file that contains the script. Alternatively, only download the install zip file, unzip, and copy the bodhi-apps-scripts folder.
4. In the Download window, ensure the files have the correct names and rename if necessary.
5. Copy the scripts package bodhi-apps-scripts.zip from Downloads to a clean folder. Unzip bodhi-apps-scripts[version].zip. Copy the install .zip from Downloads to the unzipped folder where bodhi-apps.ps1 script is.

**Note:** Ensure you download the correct .zip file for the target framework language and OS.

For example, make sure that it contains the sub-string `Linux-ARM64-dotnet` to install the dotnet suite of bodhi applications to a Raspberry Pi (ARM)  
or, ‘Linux-X64-nodejs’ to install the nodejs suite of bodhi applications to an Ubuntu x64 (Intel/AMD)

**Running the PowerShell Script:**

1. Open a cmd window or PowerShell terminal.
2. Change directory to the folder with the files.
3. Run the PowerShell script with the required parameter -targetMachine.

.\bodhi-apps.ps1 -targetMachine 192.168.0.2  
Note: Only one parameter is required for deployment, the target machine. There are many other parameters to customize the installation process, see the examples.

**What the PowerShell Script Does:**

* Asks for credentials to access the target machine over SSH.
* Copies the install .zip and bash scripts to the target.
* Runs the bash script on the target to unzip, configure, create the database and start the applications.

**Accessing the Kira Application:**

Navigate to http://192.168.0.2 with any username and password Advanced.

**Examples with Comments:**

1. **Show the scripts version:**

.\bodhi-apps.ps1 -version

1. **Show the help:**

.\bodhi-apps.ps1 -help

1. **Dry Run:**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -dryRun $true

*This will check the PC for prerequisites and print the main operations without making any changes.*

1. **Passing Username for ssh:**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -targetUsername operator

*This prepopulates the username in the credentials window.*

1. **Specify an install .zip file:**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -targetUsername operator -installFilename ..\bodhi-2024-08-20-T-12-47-40-Linux-ARM64-dotnet-TEST-plate-type-model-1c590db-1542cab.zip  
*By default, if the installFilename parameter is not specified, the install script uses the .zip file present in the current directory. Use the installFilename parameter to specify the path to the install file to use.*

1. **Install the application and configure it to allow access to the application over a specific Ip address or machine name:**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -targetMachinePublicname kira.local  
*Navigate to* <http://kira.local>

1. **Configure the application to allow access over a specific Ip address or machine name:**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -installApplications $false -targetMachinePublicname kira.local – *updateTargetMachinePublicname* $true   
*Notice the -installApplications $false, this way only the configuration is changed, and the rest of the application is left as is. Navigate to* <http://kira.local>

1. **Deploy NodeJs Suite:**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -frameworkLanguage nodejs

1. **Install the application and reinitialize the application database :**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -executeDatabaseOps reinitialize

*This deletes the old database, creates a new one, and populates it and then installs the bodhi applications.*

*Possible database operations are delete, recreate, initialize, reinitialize, migrate, seed*

1. **Database Changes Only:**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -installApplications $false -executeDatabaseOps reinitialize

*Notice the -installApplications $false, this way only the application database is changed, and the rest of the application is left as is.*

*Possible database operations are delete, recreate, initialize, reinitialize, migrate, seed*

1. **Get a Service Report:**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -getServiceReport $true

Once the report was successfully generated, please see the file in the Downloads folder. If the report .zip file is missing, please check your browser Recent downloads history for blocked downloads.

1. **Create and download an application database backup**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -getDatabaseBackup $true

Once the backup was successfully generated, please see the file in the Downloads folder. If the backup .zip file is missing, please check your browser Recent downloads history for blocked downloads.

1. **Restore the application database from a backup file**.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -restoreDatabaseFromBackupFilename c:\bodhi\database-backups\bodhi\_db\_backup\_2024-10-21\_13-06-26.zip
2. **Reinitialize all the Bodhi applications on the Target Machine**.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -reinitializeInstrument $true

Note: this is not a OS reboot, the Bodhi applications, PyRunner and PostgreSQL are restarted. The Database is not affected.

1. **Get all Protocols**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -getProtocols $true

Note: the ‘get protocols’ feature is a helper to get Revvity Preset Protocols that will be used for database seeding

1. **Get protocols**.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -getProtocols $true -getProtocolsFilter "description LIKE '%Revvity%’”
2. **Update protocols**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -targetUsername operator -updateProtocols $true -updateProtocolsFilter "name = 'MyProtocol' AND is\_current = true AND deleted\_at IS NULL" -updateProtocolsStatement " is\_current = false, last\_modified\_at = NOW(), last\_modified\_by = 'Revvity support'"

In this example, the command sets the protocol to NOT current, in effect removing it from the Protocols page. The previous Results for this protocol are still available.

*WARNING: This command works directly with the database, it is important to do a database backup first and re-check the Filter [ where ] and Statement [ set ] parameters for correctness.*

1. **Cleanup Target Machine:**

.\bodhi-apps.ps1 -targetMachine 192.168.0.2 -deleteAll $true

*WARNING: This deletes all Bodhi and PyRunner files, services, database, and logs from the target machine.*