**Configuration for My Work Laptop**

**Installation List**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Components | Verified  Installed | Todo |
| Java | Download Link  [OpenJDK Download - Corretto - AWS (amazon.com)](https://aws.amazon.com/corretto/?filtered-posts.sort-by=item.additionalFields.createdDate&filtered-posts.sort-order=desc) | XX  3/6/2024 |  |
| Installed Version  amazon-corretto-17.0.10.7.1-windows-x64-jdk | XX  3/6/2024 |  |
| JAVA\_HOME  C:\Users\myhome\JDK\amazon-corretto-17.0.10.7.1-windows-x64-jdk\jdk17.0.10\_7 | XX  3/6/2024 |  |
| java --version  openjdk 17.0.10 2024-01-16 LTS | XX  3/6/2024 |  |
| script | XX | xx |
| Maven | Download Link  [Maven – Welcome to Apache Maven](https://maven.apache.org/)  [Maven – Download Apache Maven](https://maven.apache.org/download.cgi) | XX  3/6/2024 |  |
| Installed Version  apache-maven-3.9.6-bin | XX  3/6/2024 |  |
| MVN\_HONE  C:\Users\myhome\JDK\apache-maven-3.9.6 | XX  3/6/2024 |  |
| script | XX | xx |
| Add Artifactory to Maven  [Adding Artifactory to Maven - Data Engineering Team Space - Confluence](https://upmcenterprises.atlassian.net/wiki/spaces/ITS/pages/575209483/Adding+Artifactory+to+Maven) |  | xx |
| Mirth Connect  Administrator Launcher | Download Link  [mirth-downloads (nextgen.com)](https://www.nextgen.com/solutions/interoperability/mirth-integration-engine/mirth-connect-downloads#:~:text=Administrator%20Launcher)  https://s3.amazonaws.com/downloads.mirthcorp.com/connect-client-launcher/mirth-administrator-launcher-latest-windows-x64.exe | XX  3/6/2024 |  |
| Installed Version  mirth-administrator-launcher-latest-windows-x64.exe (1.4.1) | XX  3/6/2024 |  |
| IDE | VS Code  VSCodeUserSetup-x64-1.86.2 | XX  2/22/2024 | xx |
| Java IDE | Download Link  [IntelliJ IDEA – the Leading Java and Kotlin IDE (jetbrains.com)](https://www.jetbrains.com/idea/)  [Download IntelliJ IDEA – The Leading Java and Kotlin IDE (jetbrains.com)](https://www.jetbrains.com/idea/download/?section=windows)  [Installing Cascadia Code · microsoft/cascadia-code Wiki · GitHub](https://github.com/microsoft/cascadia-code/wiki/Installing-Cascadia-Code#setting-cascadia-code-in-intellij-ide-2019) | XX  2/22/2024 |  |
| Installed Version  IntelliJ Community ideaIC-2023.3.4 | XX  2/22/2024 |  |
| WinGet ID: ⚠ JetBrains.IntelliJIDEA.Community |  |  |
| Plug in | XX | xx |
| Python | Pyenv manages Python versions and plays nicely with the pipenv dependency manager.  Uninstall all installed Pythons, and verify by   * Python –version | XX  3/7/2024 |  |
| Installing PyEnv (I choose PowerShell)  [pyenv for Windows](https://github.com/pyenv-win/pyenv-win?tab=readme-ov-file#installation)   * pyenv –version * pyenv install -l * pyenv install 3.9.13 * pyenv install 3.10.11 * pyenv install 3.11.5 * pyenv global 3.11.5 * pyenv version * python -c "import sys; print(sys.executable)" | XX  3/7/2024 |  |
| Script: Install pyenv PowerShell  Script: pyenv-win commands  Script: install python  Note: Using PyEnv | XX  3/7/2024 |  |
| Python IDE | Download Link:  [PyCharm: The Python IDE for data science and web development by JetBrains](https://www.jetbrains.com/pycharm/#:~:text=PyCharm%20Community%20Edition)  Do not install Python from WinGet or the Windows Store. | XX  3/7/2024 |  |
| Installed Version  pycharm-community-2023.3.3 | XX  2/22/2024 |  |
| Note: Plugin List | XX  3/7/2024 |  |
| Tool-Specific Fixes | [Checkov](https://bridgecrew.io/checkov/) | XX  3/7/2024 |  |
| Note: Checkov |  |  |
|  |  |  |  |
| Git | Download Link:  [Git – For Window](https://git-scm.com/download/win)  WinGet ID: Git.Git   * winget install Git.Git | XX  3/7/2024 |  |
| Installed Version  Git-2.43.0-64-bit | XX  3/5/2024 |  |
| Generate SSH Key  Open command window of Git Bash  $ ssh-keygen -t rsa -C [myhome@upmc.edu](mailto:brusyj@upmc.edu)  $ ssh-keygen -t rsa -C [brustoloni@upmc.edu](mailto:brustoloni@upmc.edu) |  |
| Postman | Installation Link  [Download Postman | Get Started for Free](https://www.postman.com/downloads/)  WinGet ID: Postman.Postman   * winget install Postman.Postman   Postman-win64-Setup | XX  3/5/2024 |  |
| Slack | [Windows | Downloads | Slack](https://slack.com/downloads/windows)  WinGet ID: SlackTechnologies.Slack   * winget install SlackTechnologies.Slack | XX  3/8/2024 |  |
|  | [Slack - Channel: general - UPMC Enterprises](https://app.slack.com/client/T0U9N2UFP/C0U9S0Y0G) |  |  |
| Terminal | [Windows Terminal - Microsoft Apps](https://apps.microsoft.com/detail/9n0dx20hk701?rtc=1&hl=en-us&gl=US) | XX  3/8/2024 |  |
|  | Windows Terminal Installer |  |
| TwistLock CLI  (twistcli) | [Prisma Cloud (Twistlock) CI/CD Twistcli - UPMCE Cloud Services](https://upmcenterprises.atlassian.net/wiki/spaces/CKB/pages/31917285/Prisma+Cloud+Twistlock+CI+CD+Twistcli) |  |  |
|  | twistcli.exe  Copy to c:/Users/myhome/bin |  |  |
| Docker Desktop | Download Link  [Docker Desktop: The #1 Containerization Tool for Developers | Docker](https://www.docker.com/products/docker-desktop/) | XX  3/5/2024 |  |
| Installed Version  Docker Desktop Installer (downloaded 3/5/2024) | XX  3/5/2024 |  |
| NodeJS | Pre-requirement: **Allow Users to Create Symbolic Links** |  |  |
| Download Link  [Releases · coreybutler/nvm-windows (github.com)](https://github.com/coreybutler/nvm-windows/releases)  [Command-line API | Node.js v20.11.1 Documentation (nodejs.org)](https://nodejs.org/dist/latest-v20.x/docs/api/cli.html) | XX  3/6/2024 |  |
| Installed version.  nvm-setup (3/6/2024 4:23 PM)  Choose Installed Home:  C:\Users\myhome\nvm  Created environment variable:  NVM\_HOME = %USERPROFILE%\nvm  NVM\_SYMLINK = C:\Program Files\nodejs | XX  3/6/2024 |  |
| Install npm and node.js  > nvm install lts  > nvm use lts  Result: node.js version 20.11.1 (64-bit) | XX  3/6/2024 |  |
| script | XX |  |
| Helm  [Helm](https://helm.sh/) | Requires: Docker  [Installing Helm](https://helm.sh/docs/intro/install/)  [Releases · helm](https://github.com/helm/helm/releases) | XX  3/7/2024 |  |
| Installed Version:  helm-v3.14.2-windows-amd64.zip  The Helm is a self-contained executable.  I recommend just copying it to %USERPROFILE%\bin  and ensuring that the directory is in the User's PATH.  Copied helm.exe to C:\Users\myhome\bin  Add C:\Users\myhome\bin to path | XX  3/7/2024 |  |
|  | [Deploying a New Source with helm-](https://upmcenterprises.atlassian.net/wiki/spaces/ITS/pages/1012662496/Deploying+a+New+Source#Adding-the-Helm-Repo) |  |  |
| Script: Helm |  |  |
| [HL7Spy](https://hl7spy.ca/) | [HL7 Spy – License Info](https://upmcenterprises.atlassian.net/wiki/spaces/ITS/pages/1307869789/HL7+Spy) | XX  3/7/2024 |  |
|  | Download Link  [HL7Spy Download](http://ihs.48thave.com/blog/) | XX  3/7/2024 |  |
|  | Version HL7Spy - v2.3.604.0 - Professional  Email/Licensee:  [1UPMC-ACCT@upmc.org](mailto:1UPMC-ACCT@upmc.org)  License Key:  FAatgtDSpDzn5WCjP5u2o4x7QlaOIfc+ITIKbtIp1bXQ6DfJQrGkTaQgHIeHS4/L2vGN7oHA0Hpj1Y7JZxhGTA=2PW130811 | XX  3/7/2024 |  |
| KeePass | Installation Link  [Downloads - KeePass](https://keepass.info/download.html) (We use KeePass 2.x)  WinGet ID: DominikReichl.KeePass   * winget install DominikReichl.KeePass | XX  3/7/2024 |  |
| [KeePass Password Manager -Set Up Syncronization](https://upmcenterprises.atlassian.net/wiki/spaces/ITS/pages/2168750090/KeePass+Password+Manager) | XX  3/7/2024 |  |
| Kubernetes CLI  - Kubectl | Installation Link:  [Install and Set Up kubectl on Windows | Kubernetes](https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/) | XX  3/7/2024 |  |
| Installed version 1.29.1  Copy kubectl.exe to C:\Users\myhome\bin   * kubctl version | XX  3/7/2024 |  |
| Script: Kubectl | XX  3/7/2024 |  |
| OpenShift CLI  (oc) | The OpenShift CLI is a self-contained executable. I recommend just copying it to %USERPROFILE%\bin and ensuring that the directory is in the User's PATH.  Installation Link:  Screenshot: OC- Install (below)  To get download file oc.zip | XX  3/8/2024 |  |
| Installed Version:  Get oc.exe from oc.zip  Copied to c:\User\myhome\bin | XX  3/8/2024 |  |
| Script: download oc.zip | XX  3/8/2024 |  |
| OhMyPosh | PowerShell Prompt enhancements  [Windows | Oh My Posh](https://ohmyposh.dev/docs/installation/windows) |  |  |
| AWS Tools | | | |
| AWS CDK | Pre-requirement: NodeJS |  |  |
| Installed Version   * npm install -g aws-cdk   Results: New minor version of npm available! 10.2.4 -> 10.5.0 | XX  3/6/2024 |  |
| AWS CLI | Download Links  [Command Line Interface - AWS CLI - AWS (amazon.com)](https://aws.amazon.com/cli/)  Or  WinGet ID: Amazon.AWSCLI   * winget install Amazon.AWSCLI | XX  3/6/2024 |  |
| Installed Version  AWS CLI v2 | XX  3/6/2024 |  |
| Account setup:  Ensure that the AWS\_CA\_BUNDLE **Environment Variables** is set. | XX  3/6/2024 |  |
| AWS SSO Setup  [AWS SSO Setup](https://upmcenterprises.atlassian.net/wiki/spaces/ITS/pages/1402306583/AWS+SSO+Setup)  [IAM Identity Center (successor to AWS Single Sign-On (SSO)) Standard](https://upmcenterprises.atlassian.net/wiki/spaces/CKB/pages/526386284/IAM+Identity+Center+successor+to+AWS+Single+Sign-On+SSO+Standard) | XX  3/6/2024 |  |
| AWS Session Manager Plug-in | Note: install Session Manager Plug-in is necessary for SSH to AWS bastion to work  [Install the Session Manager plugin on Windows](https://docs.aws.amazon.com/systems-manager/latest/userguide/install-plugin-windows.html)  >winget install Amazon.SessionManagerPlugin | XX  3/18/2024 |  |
| Installed Version  SessionManagerPluginSetup (File date: 3/18/2024)  Add C:\Program Files\Amazon\SessionManagerPlugin\bin to path |  |
| PostgreSQL  Database Management | Installed Version  postgresql-16.2-1-windows-x64 | XX  4/10/2024 |  |
| Matt Setup Project in GIT: | [Matt Himrod / workstation-setup · GitLab (upmc.com)](https://gitlab.upmc.com/himrodma/workstation-setup) |  |  |

**Configuration**

**Allow Users to Create Symbolic Links**

By default, Windows requires Administrator rights to create symbolic links. This can be changed in the Local Security Policy.

* Open the Local Security Policy management console as Administrator
* Find the following path in the heirarchy on the left: Security Settings > Local Policies > User Rights Assignment
* In the policy window on the right, find "Create Symbolic Links".
* Double-click the policy. Click "Add User or Group". Add your user.

**Cascadia Fonts**

* Download the latest font release zip file: [Releases · microsoft/cascadia-code (github.com)](https://github.com/microsoft/cascadia-code/releases)
* Extract the contents of the zip file.
* Within the ttf directory, right-click on each font file and click "Install".
* Set up the font in your IDE.

[Installing Cascadia Code · microsoft/cascadia-code Wiki](https://github.com/microsoft/cascadia-code/wiki/Installing-Cascadia-Code#installing-cascadia-code-in-vs-code)

**Environment Variables**

Each section will describe any necessary environment variables for the tool set being installed. Here is how one sets these varialbes in the user context:

Using the GUI Tool:

* C:\Windows\System32\rundll32.exe sysdm.cpl,EditEnvironmentVariables

Using PowerShell:

[System.Environment]::SetEnvironmentVariable('Variable Name', 'Variable Value', 'User')

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value | Used By | Description |
| ACTIVE\_PROFILE | dev | SpringBoot | Determines property files to load, based on environment |
| \*AWS\_CA\_BUNDLE | %USERPROFILE%  /.certs/AWS\_AND\_UPMC\_BUNDLE.pem | OpenSSL AWS CLI | UPMC Certificate bundle for SSL necessary for the UPMC Corporate Network. |
| MAVEN\_REPO\_PASS | <Artifactory API Key>  Reference .env | Maven Python | Used to authenticate to Artifactory repositories. |
| MAVEN\_REPO\_USER | <UPMC Email Address>  myhome@upmc.edu | Maven Python | Used to authenticate to Artifactory repositories. |
| \*NODE\_EXTRA\_CA\_CERTS | %USERPROFILE%  /.certs/AWS\_AND\_UPMC\_BUNDLE.pem | OpenSSL NodeJS 18+ | UPMC Certificate bundle for SSL necessary for the UPMC Corporate Network. |
| \*NODE\_OPTIONS | --openssl-config  =  %USERPROFILE%/.conf/upmc\_openssl.conf | OpenSSL 3.0+ NodeJS 18+ | OpenSSL Config file with settings necessary for the UPMC Corporate Network. |
| NVM\_HOME | %USERPROFILE%/nvm | NVM | Required NVM Variable. |
| NVM\_SYMLINK | C:/Program Files/nodejs | NVM | Required NVM Variable. |
| \*OPENSSL\_CONF | %USERPROFILE%/.conf/upmc\_openssl.conf | OpenSSL 3.0+ NodeJS 18+ Python 3.11.6+ | OpenSSL Config file with settings necessary for the UPMC Corporate Network. |
| PIPENV\_VENV\_IN\_PROJECT | 1 | Pipenv | Directs Pipenv to create Virtual Environments in the project directory, not the user profile directory. |
| \*PYENV | %USERPROFILE%/.pyenv/pyenv-win/ | Pyenv | Required Pyenv variable. |
| \*PYENV\_HOME | %USERPROFILE%/.pyenv/pyenv-win/ | Pyenv | Required Pyenv variable. |
| \*PYENV\_ROOT | %USERPROFILE%/.pyenv/pyenv-win/ | Pyenv | Required Pyenv variable. |

* Need to set under user variables

**OpenSSL 3.0 Environment Variables**

[/docs/man3.0/man7/openssl-env.html](https://www.openssl.org/docs/man3.0/man7/openssl-env.html)

$EnvFromReg = ([Microsoft.Win32.Registry]::CurrentUser.OpenSubKey('Environment', $true))

$EnvFromReg.SetValue('OPENSSL\_CONF', '%USERPROFILE%\.certs\upmc\_openssl.conf', "ExpandString")

**OpenSSH Authentication Key**

You will need an OpenSSH key for authentication. For compatibility with AWS and Azure SSH endpoints, we will need to use the RSA algorithm. When complying with FIPS 186-2, DSA keys must be 1024 bits, which does not offer sufficient security. RSA 2048 is sufficient, but RSA 4096 is best.

``PowerShell ssh-keygen.exe -b 4096 -t rsa

|  |  |
| --- | --- |
| If the PuTTY Suite is available, the following will generate a very strong key:  ```PowerShell | |
|  | puttygen.exe -b 4096 -t rsa --primes proven --strong-rsa |

**SSH Configuration using AWS Session Manager**

**UPMC Certificate Bundles**

You will need to obtain a copy of the UPMC CA Certificate from one of the following locations:

I recommend saving these files in %USERPROFILE%\.certs (a .certs folder under your user profile.)

Copy [upmc\_openssl.conf](upmc\_openssl.conf) to %USERPROFILE%\.certs.

**Scripts**

|  |  |
| --- | --- |
| JDK Script: | |
|  | New-Item -Type Directory ~\JDK  Set-Location ~\JDK  Invoke-RestMethod -Uri https://corretto.aws/downloads/latest/amazon-corretto-21-x64-windows-jdk.zip -OutFile amazon-corretto-jdk.zip  Invoke-RestMethod -Uri https://corretto.aws/downloads/latest\_sha256/amazon-corretto-21-x64-windows-jdk.zip -OutFile amazon-corretto-sha256  if ((Get-FileHash .\amazon-corretto-jdk.zip ).Hash -eq (Get-Content .\amazon-corretto-sha256)) {  Expand-Archive amazon-corretto-jdk.zip -DestinationPath .  # This line only works when there is one JDK in ~\JDK  $EnvFromReg = ([Microsoft.Win32.Registry]::CurrentUser.OpenSubKey('Environment', $true))  $EnvFromReg.SetValue('JAVA\_HOME', (Get-ChildItem -Directory jdk\*).FullName, "ExpandString")  $userPath = $EnvFromReg.GetValue('Path', $null, "DoNotExpandEnvironmentNames") -split ';'  $userPath += '%JAVA\_HOME%\bin'  $EnvFromReg.SetValue('Path', $userPath.Where({$\_}) -Join ';', "ExpandString")  Remove-Item amazon-corretto-jdk.zip, amazon-corretto-sha256  } else {  Write-Host -ForegroundColor Red 'Hash validation failed.'  } |

|  |  |
| --- | --- |
| IntelliJ IDEA Plug Ins | |
| IntelliJ IDEA | * CheckStyle-IDEA * Docker * EnvFile * Handlebars/Mustache * Lombok * SonarLint * Build Tools: Maven * Code Coverage: Code Coverage for Java * HTML and XML: Html Tools * IDE Settings: Settings Sync * JVM Frameworks: JavaFX * Languages: Markdown * Languages: Properties * Languages: Shell Script * Languages: Toml * Languages: YAML * Test Tools: JUnit * Version Controls: Git *  Version Controls: GitLab |

|  |  |
| --- | --- |
| Maven Script: | |
|  | Set-Location ~\JDK  Invoke-RestMethod -Uri https://dlcdn.apache.org/maven/maven-3/3.9.6/binaries/apache-maven-3.9.6-bin.zip -OutFile .  Invoke-RestMethod -Uri https://downloads.apache.org/maven/maven-3/3.9.6/binaries/apache-maven-3.9.6-bin.zip.sha512 -OutFile .  if ((Get-FileHash -Algorithm SHA512 .\apache-maven-3.9.6-bin.zip).Hash -eq (Get-Content .\apache-maven-3.9.6-bin.zip.sha512)) {  Expand-Archive apache-maven-3.9.6-bin.zip -DestinationPath .  [System.Environment]::SetEnvironmentVariable('MVN\_HOME', '%USERPROFILE%\JDK\apache-maven-3.9.6', 'User')  $EnvFromReg = ([Microsoft.Win32.Registry]::CurrentUser.OpenSubKey('Environment', $true))  $EnvFromReg.SetValue('MVN\_HOME', (Get-ChildItem -Directory jdk\*).FullName, "ExpandString")  $userPath = $EnvFromReg.GetValue('Path', $null, "DoNotExpandEnvironmentNames") -split ';'  $userPath += '%MVN\_HOME%\bin'  $EnvFromReg.SetValue('Path', $userPath.Where({$\_}) -Join ';', "ExpandString")  Remove-Item apache-maven-3.9.6-bin.zip, apache-maven-3.9.6-bin.zip.sha512  } else {  Write-Host -ForegroundColor Red 'Hash validation failed.'  } |
| After restarting the shell, run the following to add our custom CAs to the certificate store. This will need to be done each time a new JDK is downloaded and activated. | |
|  | Set-Location ~\.certs  keytool -import -cacerts -storepass changeit -noprompt -file UPMC-ROOT-CA.pem |

|  |  |
| --- | --- |
| NVM Script: | |
|  | New-Item -Type Directory ~\nvm  Set-Location ~\nvm  Invoke-RestMethod -Uri https://github.com/coreybutler/nvm-windows/releases/download/1.1.12/nvm-noinstall.zip -OutFile nvm-noinstall.zip  Invoke-RestMethod -Uri https://github.com/coreybutler/nvm-windows/releases/download/1.1.12/nvm-noinstall.zip.checksum.txt -OutFile nvm-noinstall.zip.checksum.txt  if ((Get-FileHash nvm-noinstall.zip -Algorithm MD5).Hash.ToLower() -eq (Get-Content nvm-noinstall.zip.checksum.txt)) {  Expand-Archive nvm-noinstall.zip -DestinationPath .  $EnvFromReg = ([Microsoft.Win32.Registry]::CurrentUser.OpenSubKey('Environment', $true))  $EnvFromReg.SetValue('NVM\_HOME', "%USERPROFILE%\nvm", "ExpandString")  $EnvFromReg.SetValue('NVM\_SYMLINK', "%USERPROFILE%\nodejs", "ExpandString")    $userPath = $EnvFromReg.GetValue('Path', $null, "DoNotExpandEnvironmentNames") -split ';'  $userPath += '%NVM\_HOME%;%NVM\_SYMLINK%'  $EnvFromReg.SetValue('Path', $userPath.Where({$\_}) -Join ';', "ExpandString")  $EnvFromReg.SetValue('NODE\_EXTRA\_CA\_CERTS', '%USERPROFILE%\.certs\AWS\_AND\_UPMC\_BUNDLE.pem', "ExpandString")  $EnvFromReg.SetValue('NODE\_OPTIONS', '--openssl-config=%USERPROFILE%\.certs\upmc\_openssl.conf', "ExpandString")  Write-Output "root: $(Join-Path $ENV:USERPROFILE 'nvm')" > settings.txt  Write-Output "path: $(Join-Path $ENV:USERPROFILE 'nodejs')" >> settings.txt  Write-Output "proxy: none" >> settings.txt  Write-Output "arch: 64" >> settings.txt  Remove-Item nvm-noinstall.zip, nvm-noinstall.zip.checksum.txt  } else {  Write-Host -ForegroundColor Red 'Hash validation failed.'  } |

|  |  |
| --- | --- |
| AWS\_CA\_BUNDLE [environment variable](/himrodma/workstation-setup/-/blob/main/Configuration.md#environment-variables) is set. | |
|  | $EnvFromReg = ([Microsoft.Win32.Registry]::CurrentUser.OpenSubKey('Environment', $true))  $EnvFromReg.SetValue('AWS\_CA\_BUNDLE', '%USERPROFILE%\.certs\AWS\_AND\_UPMC\_BUNDLE.pem', "ExpandString") |

|  |  |
| --- | --- |
| pyenv-win commands | |
|  | commands List all available pyenv commands  local Set or show the local application-specific Python version  global Set or show the global Python version  shell Set or show the shell-specific Python version  install Install 1 or more versions of Python  uninstall Uninstall 1 or more versions of Python  update Update the cached version DB  rehash Rehash pyenv shims (run this after switching Python versions)  vname Show the current Python version  version Show the current Python version and its origin  version-name Show the current Python version  versions List all Python versions available to pyenv  exec Runs an executable by first preparing PATH so that the selected  Python version's `bin' directory is at the front  which Display the full path to an executable  whence List all Python versions that contain the given executable |

|  |  |
| --- | --- |
| Install pyenv PowerShell Script: | |
|  | Invoke-WebRequest -UseBasicParsing -Uri "https://raw.githubusercontent.com/pyenv-win/pyenv-win/master/pyenv-win/install-pyenv-win.ps1" -OutFile "./install-pyenv-win.ps1"; &"./install-pyenv-win.ps1" |

|  |  |
| --- | --- |
| After installing Pyenv, install Python versions, update pip, install pipenv, and set a default global version: | |
|  | @('3.9.13','3.10.11','3.11.7','3.12.1') | % { pyenv install $\_; pyenv local $\_; python -m pip install --upgrade pip pipenv; pyenv rehash }  pyenv local --unset  pyenv global 3.11.7  [System.Environment]::SetEnvironmentVariable('PIPENV\_VENV\_IN\_PROJECT', '1', 'User') |

|  |  |
| --- | --- |
| Note: Using PyEnv | |
|  | The pyenv rehash command will create/refresh shims that are placed in the system path. These shims determine the Python attached to the current directory or the global Python if none is set and redirect execution to the proper version. Some scripts expect a standalone Python installation and will not work correctly. For certain scripts, running pyenv rehash will cause pyenv to create the necessary shims for the command to work. If this doesn't work, using pyenv exec in front of the command will work for the majority of the remaining cases. Certain scripts may require further workarounds. |

|  |  |
| --- | --- |
| Note: PyCharm Plugin List (Installed) | |
|  | * Docker * EnvFile * Handlebars/Mustache * SonarLint * IDE Settings: Settings Sync * Languages: Markdown * Languages: Properties * Languages: Shell Script * Languages: Toml * Languages: YAML * Test Tools: JUnit * Version Controls: Git * Version Controls: GitLab |

|  |  |
| --- | --- |
| Note: Checkov | |
|  | Checkov is an open source policy-as-code tool that scans for security issues in infrastructure as code (IaC) templates, container images, and pipeline configuration. We use it to scan the CloudFormation templates produced by our CDK scripts.  Checkov does not work well with pyenv. The Checkov launch script works by finding the application associated with .py files and then re-launching . This may be unset or it may launch a different version of Python than intended -- one which does not have checkov installed.  function run\_checkov {  param (  [Parameter(Position = 0, ValueFromRemainingArguments)]  [String[]]$Parameters  )  $checkov\_cmd = Get-Command \* | Where-Object Name -Like 'checkov\*'  if ($checkov\_cmd) {  & python -x $($checkov\_cmd.Path) $Parameters  } else {  Write-Host -ForegroundColor Red 'Unable to locate checkov.cmd (Is your .venv active?)'  }  } |
|  |  |

|  |  |
| --- | --- |
| Script:helm | |
|  | Invoke-RestMethod -Uri https://get.helm.sh/helm-v3.14.0-windows-amd64.zip -OutFile .  Invoke-RestMethod -Uri https://get.helm.sh/helm-v3.14.0-windows-amd64.zip.sha256sum -OutFile .  if ((Get-FileHash .\helm-v3.14.0-windows-amd64.zip).Hash -eq (Get-Content .\helm-v3.14.0-windows-amd64.zip.sha256sum -Delimiter ' ' -TotalCount 1)) {  Expand-Archive helm-v3.14.0-windows-amd64.zip  Get-ChildItem -Recurse -Filter helm.exe | % { Move-Item $\_ .}  Remove-Item helm-v3.14.0-windows-amd64.zip, helm-v3.14.0-windows-amd64.zip.sha256sum, .\helm-\*-windows-amd64 -Recurse  } else {  Write-Host -ForegroundColor Red 'Hash validation failed.'  } |

|  |  |
| --- | --- |
| Script: Kubectl | |
|  | Invoke-RestMethod -Uri https://dl.k8s.io/release/v1.29.1/bin/windows/amd64/kubectl.exe -OutFile .  Invoke-RestMethod -Uri https://dl.k8s.io/v1.29.1/bin/windows/amd64/kubectl.exe.sha256 -OutFile .  if ((Get-FileHash .\kubectl.exe).Hash -eq (Get-Content .\kubectl.exe.sha256)) {  Remove-Item kubectl.exe.sha256  } else {  Rename-Item kubectl.exe kubectl.exe.bad  Write-Host -ForegroundColor Red 'Hash validation failed.'  } |

Screenshot: OC- Install

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| Script: download oc.zip | |
|  | Invoke-RestMethod -Uri https://downloads-openshift-console.apps.cpd.isd.upmc.edu/amd64/windows/oc.zip -OutFile .  Expand-Archive oc.zip  Remove-Item oc.zip |

|  |  |
| --- | --- |
| XXX Script: | |
|  |  |

|  |  |
| --- | --- |
| XXX Script: | |
|  |  |