CyberRes

Cloud DevSecOps

Application Security

Rohit Baryha, CyberRes Market Development Technical Consultant

Fortify Portfolio

Automate testing throughout the CI/CD pipeline, enable developers to quickly resolve issues

- **Static Code Analyzer (SCA):** Analyzes source code for security vulnerabilities (SAST)
- **WebInspect:** Dynamic testing (DAST) analyzes applications in their running state and simulates attacks against an application to find vulnerabilities
- **Fortify on Demand (FoD):** AppSec as a Service, that includes SAST, DAST, and MAST
- **Software Security Center:** Holistic application security platform included with on-premises solutions to get complete visibility of application security risks
- **Sonatype:** Scans open source components for vulnerabilities

	Static (SAST)	Dynamic (DAST)
On Prem	✓ *Static Code Analyzer (SCA)*	✓ *WebInspect*
As a Service	✓ *Fortify on Demand (FoD)*	✓ *Fortify on Demand (FoD)*

Solutions that Align With DevSecOps Success — Integration — Automation — Speed

Backed by the Market Leading Software Security Research Team

1,000+Vulnerability Categories | 27 Programming Languages | 1M+ Individual APIs

Demo Overview

Azure DevOps

- Integration using Build / Release Tasks
- Integration using YML / Classic UI
- Integration with Fortify On Prem
- Static Analysis Integration
- Source Code Composition analysis using Sonatype
- Dynamic Analysis using ScanCentral

AWS CodeStar

- Integration using YML
- Integration with Fortify On Prem
- Static Analysis Integration
- Source Code Composition analysis using Sonatype
- Dynamic Analysis using ScanCentral
- Continuous Feedback

GCP CloudBuild

- Integration using YML
- Integration with Fortify On Prem
- Static Analysis Integration with docker build

Pre-Requisites

Cloud DevOps – CI / CD Environment

Cloud Hosted Runner / Agent*

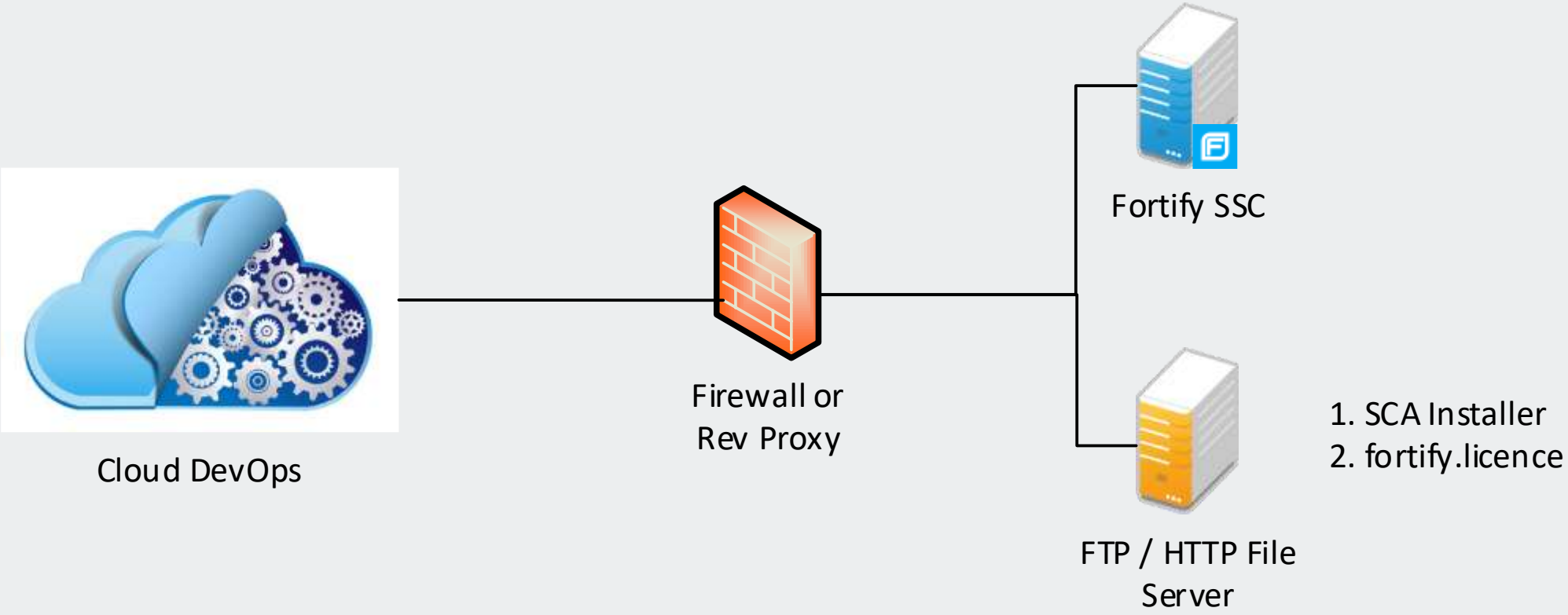
- Standard images - Linux, Windows, MacOS
- Most of the tools are pre-installed
- Can use Container images from Docker Hub*
- Runs on Cloud and Self Destroyed when pipeline stops
- Environment cannot be retained

Self Hosted Runner / Agent

- Customized & Flexible images – Linux, Windows, MacOS
- Customer has to install and configure the environment
- Cloud DevOps Agent need to be installed and configure.
- Cloud DevOps Agent and Cloud Environment has to communicate to each other.
- Environment can be retained & tailored as per requirement

* Depends on cloud vendor

Native SAST Integration – Traditional Approach



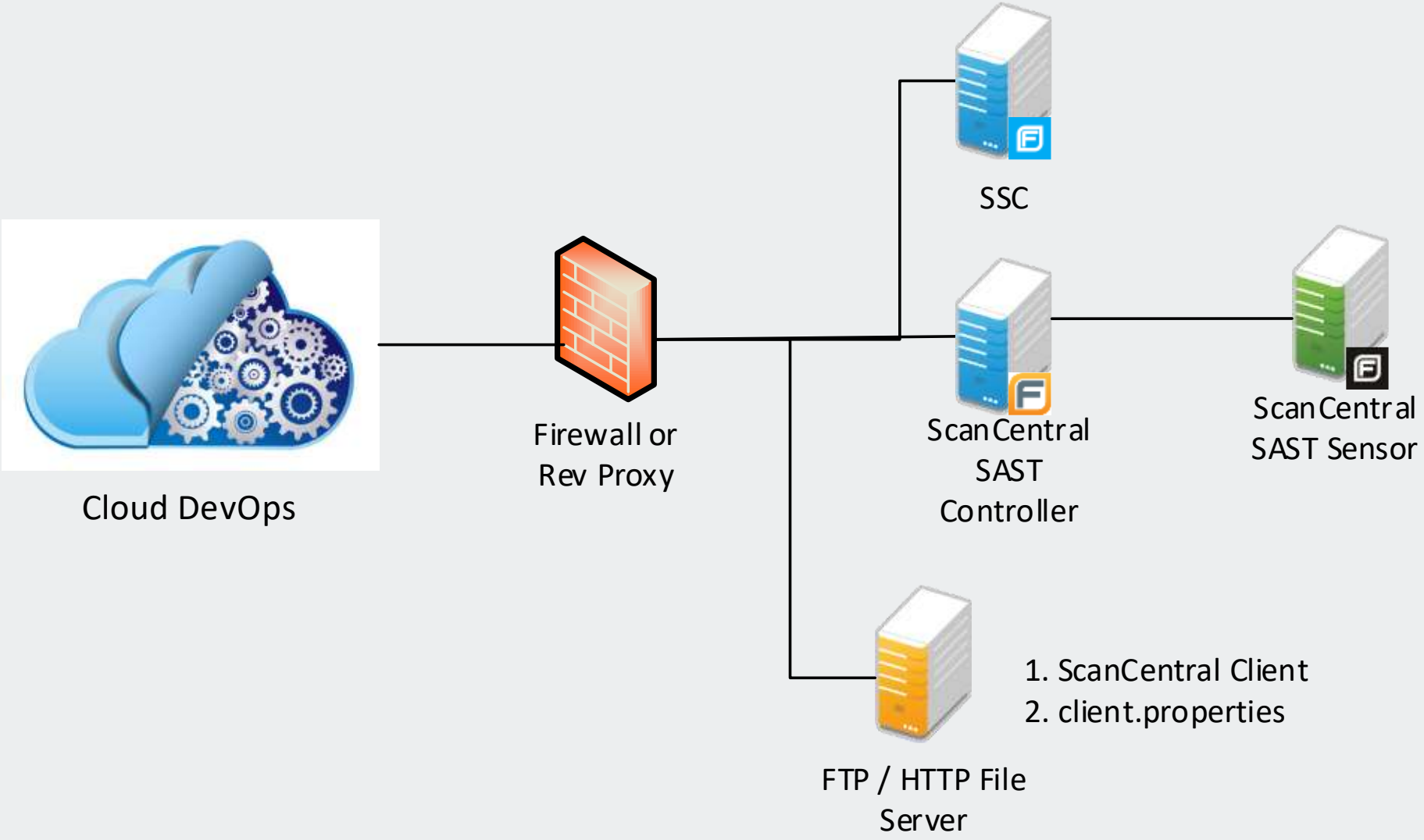
Native SAST Integration - Traditional Approach - Steps

1. Download the SCA Installer file ~1 GB
2. Download the fortify.license file
3. Install the SCA using installer and license file
4. Update the Rulepacks
5. Translate the code using SCA CLI
6. Analyze the code using SCA CLI
7. Upload the results to SSC via FortifyClient
8. Apply Quality Gate via FPRUtility (Optional)

SSC – Allow Download files

- Modify \ssc\WEB-INF\internal\securityContext.xml
- Uncomment –
  - <security:intercept-url pattern="/downloads/**" access="PERM_ANONYMOUS" />
- Copy required files in \ssc\WEB-INF\downloads or \ssc\downloads folder
  - Fortify_SCA_and_Apps_20.2.0_windows_x64.exe
  - fortify.license file
- http://ip:8180/ssc/downloads/Fortify_SCA_and_Apps_20.2.0_windows_x64.exe
- http://ip:8180/ssc/downloads/fortify.license

NextGen SAST Integration – ScanCentral SAST Approach

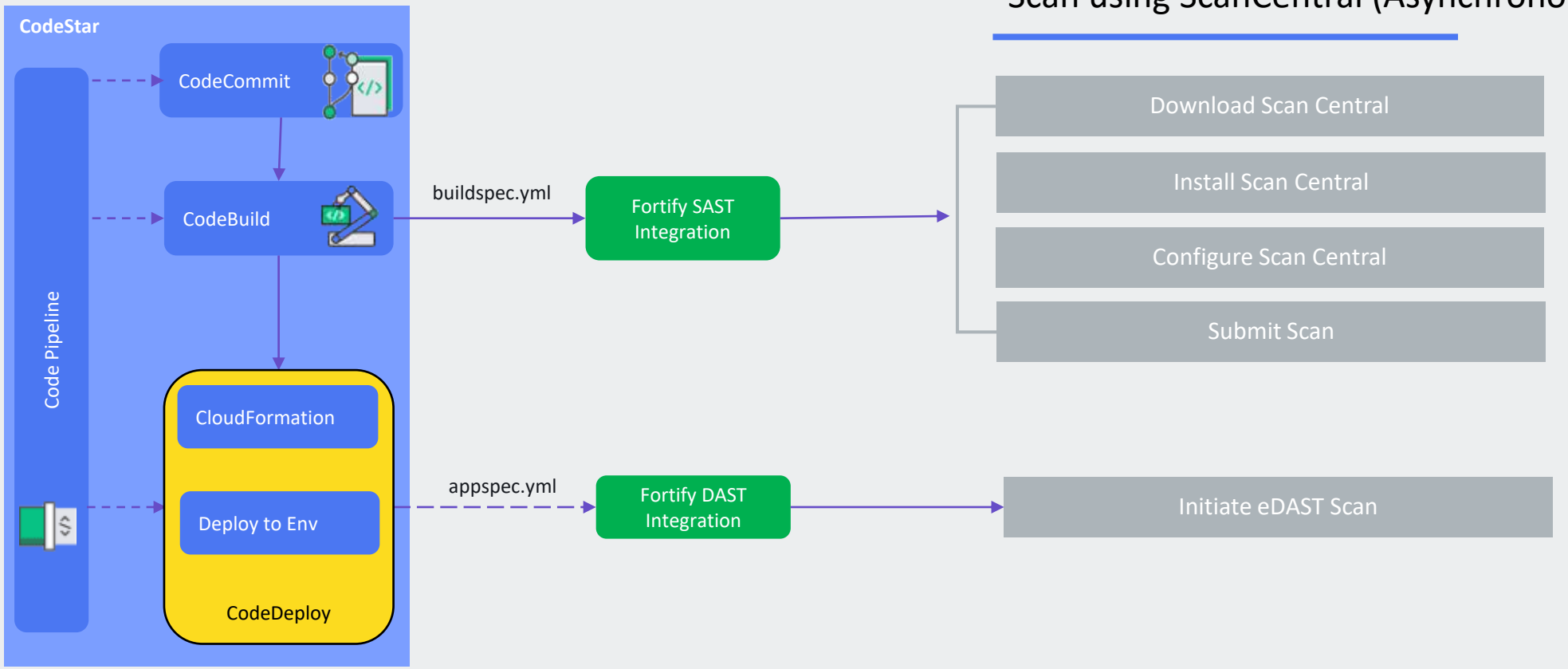


NextGen SAST Integration – ScanCentral Approach

1. Download the ScanCentral Client file ~60 MB, {required Java version >=1.8}
2. Extract the ScanCentral Client
3. Download / Create the client.properties file
4. Translate the code (conditional)
5. Upload the code via ScanCentral Client to ScanCentral Controller
6. Results will be uploaded to SSC
7. Quality Gate (may be in next release)

AWS CodeStar

CodeBuild using SCA and ScanCentral



Scan using ScanCentral (Asynchronous Process)

- Download Scan Central
- Install Scan Central
- Configure Scan Central
- Submit Scan

Initiate eDAST Scan

AWS DevOps

Components of AWS CodeStar

- AWS Module: CodeStar
- appspec.yml - this file is used by AWS CodeDeploy when deploying the web application to EC2
- **buildspec.yml** - this file is used by AWS CodeBuild to build the web application
- pom.xml - this file is the Maven Project Object Model for the web application
- src/main - this directory contains your Java service source files
- src/test - this directory contains your Java service unit test files
- scripts/ - this directory contains scripts used by AWS CodeDeploy when installing and deploying your application on the Amazon EC2 instance
- template.yml - this file contains the description of AWS resources used by AWS CloudFormation to deploy your infrastructure
- template-configuration.json - this file contains the project ARN with placeholders used for tagging resources with the project ID

AWS - buildspec.yml

DVJA / buildspec.yml Info

```yaml
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
    commands:
      # Upgrade AWS CLI to the latest version
      - pip install --upgrade awscli
  pre_build:
    commands:
      #- mvn clean compile test
      - mvn clean
  build:
    commands:
      - mvn package
  post_build:
    commands:
      # Do not remove this statement. This command is required for AWS CodeStar projects.
      # Update the AWS Partition, AWS Region, account ID and project ID in the project ARN in template-configuration.json file so AWS CloudFormation can tag project resources.
      - sed -i.bak 's/\$PARTITION\$/'${PARTITION}'/g;s/\$AWS_REGION\$/'${AWS_REGION}'/g;s/\$ACCOUNT_ID\$/'${ACCOUNT_ID}'/g;s/\$PROJECT_ID\$/'${PROJECT_ID}'/g' template-configuration.json
      - bash scascan.bash
artifacts:
  files:
    - 'appspec.yml'
    - 'template.yml'
    - 'scripts/*'
    - 'target/ROOT.war'
    - 'template-configuration.json'

```

SCAScan.bash

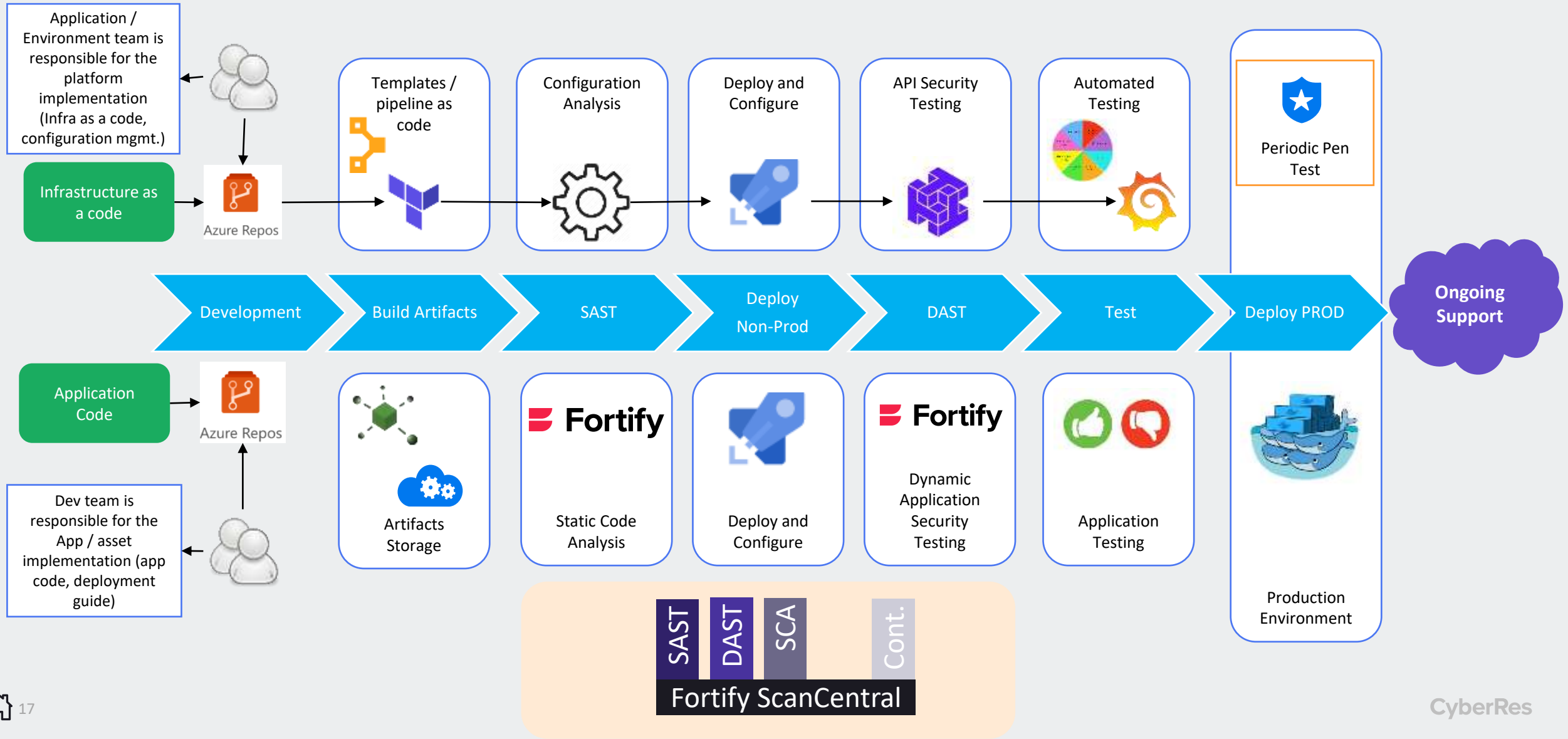
DVJA / scascan.bash Info

Edit

```
#!/bin/bash

#Getting downloaders
wget -nv --no-check-certificate https://[redacted]:8443/ssc/downloads/Fortify_SCA_and_Apps_20.2.0_linux_x64.run

wget -nv --no-check-certificate https://[redacted]:8443/ssc/downloads/fortify.license

wget --no-check-certificate https://[redacted]:8443/ssc/downloads/SSC.cer
chmod +x ./Fortify_SCA_and_Apps_20.2.0_linux_x64.run
./Fortify_SCA_and_Apps_20.2.0_linux_x64.run --fortify_license_path ./fortify.license --mode unattended --installdir
/opt/Fortify/Fortify_SCA_and_Apps_20.2.0/
/opt/Fortify/Fortify_SCA_and_Apps_20.2.0/bin/fortifyupdate
/opt/Fortify/Fortify_SCA_and_Apps_20.2.0/jre/bin/keytool -importcert -trustcacerts -alias SSC -keystore
/opt/Fortify/Fortify_SCA_and_Apps_20.2.0/jre/lib/security/cacerts -file ./SSC.cer -storepass changeit -noprompt
/opt/Fortify/Fortify_SCA_and_Apps_20.2.0/bin/sourceanalyzer -b aws-java -clean
/opt/Fortify/Fortify_SCA_and_Apps_20.2.0/bin/sourceanalyzer -b aws-java -source 1.8 -cp "./**/*.jar" "./**/*.java"
"./**/*.js" "./**/*.jsp" "./**/*.html" "./**/*.properties" "./**/*.xml"
/opt/Fortify/Fortify_SCA_and_Apps_20.2.0/bin/sourceanalyzer -b aws-java -scan -f AWS-SAST.fpr
/opt/Fortify/Fortify_SCA_and_Apps_20.2.0/bin/fortifyclient -url https://[redacted]:8443/ssc/ -authtoken 169c2c01-
8902-4b42-af2c-1b11f8536f85 -applicationVersionID "10023" uploadFPR -file AWS-SAST.fpr

```

Azure DevOps

Fortify enable Azure DevOps



Azure DevOps

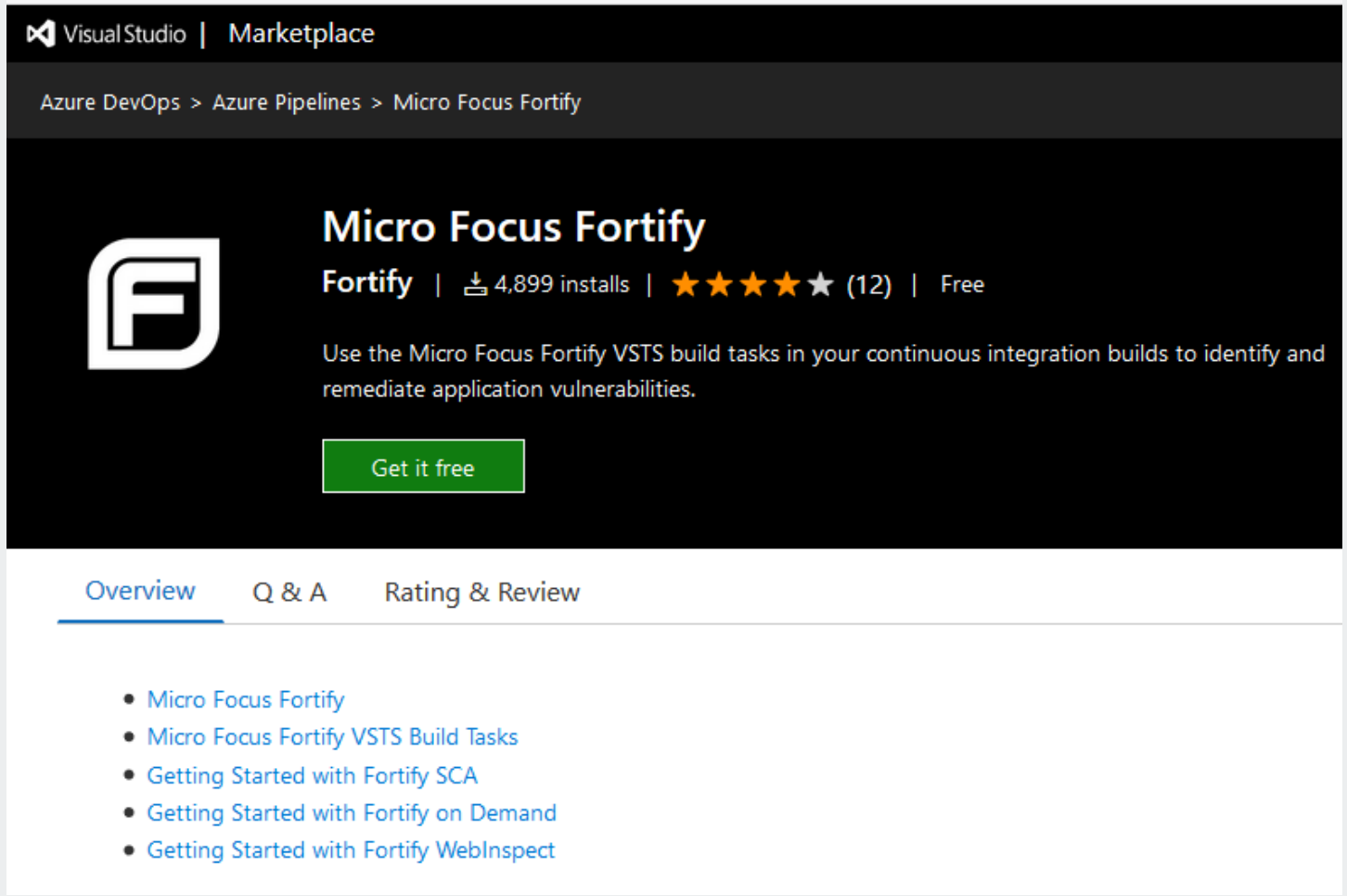
- Integration is with Build Pipeline
- Plugins are available
- Main config file is **azure-pipelines.yml** file

Azure DevOps “Free Tier” Limitation

- Public project: 10 free Microsoft-hosted parallel jobs that can run for up to 360 minutes (6 hours) each time, with no overall time limit per month.
- Private project: One free job that can run for up to 60 minutes each time, until you've used 1,800 minutes (30 hours) per month.
  - This 60 minutes limitation requires special attention on DAST Pipeline
- Ref: https://docs.microsoft.com/en-us/azure/devops/pipelines/licensing/concurrent-jobs?view=azure-devops

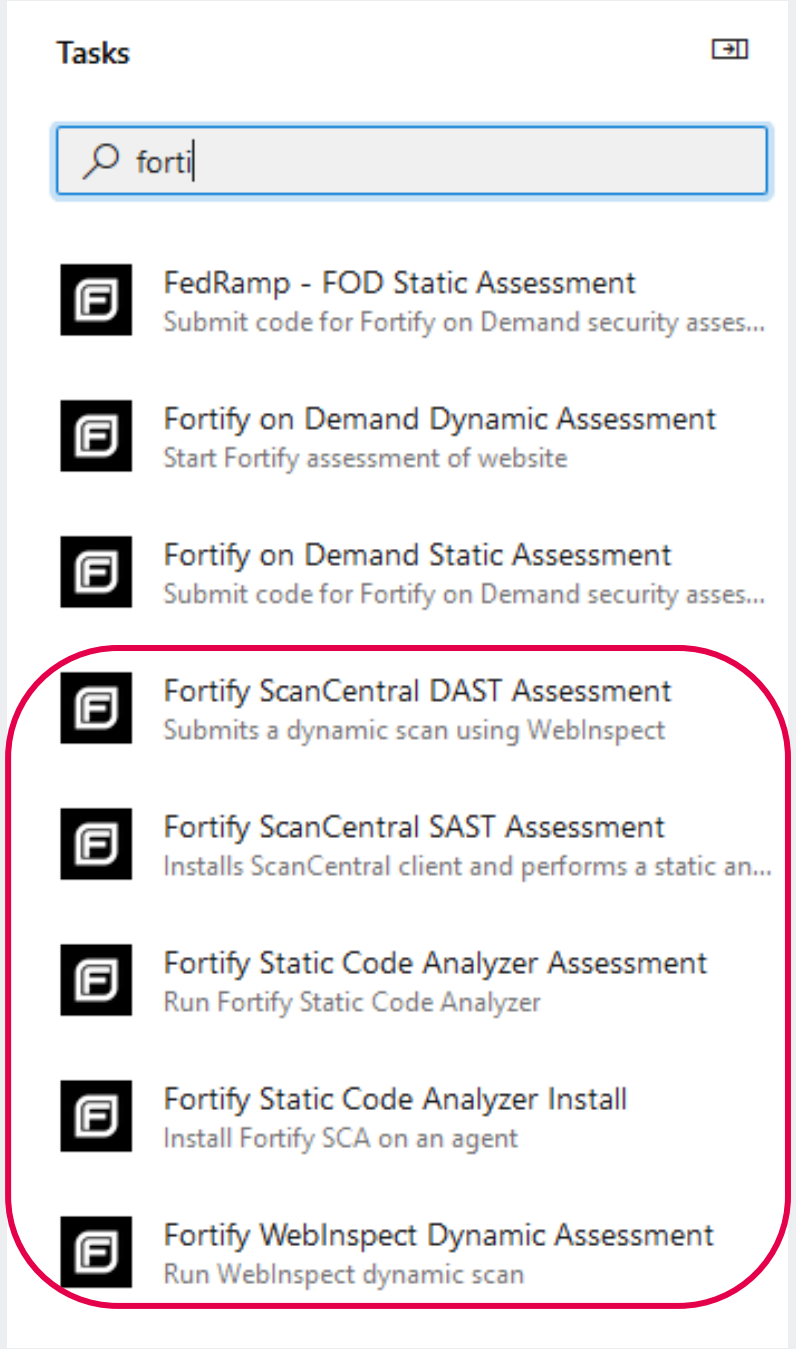
Fortify Azure DevOps Extension

- https://marketplace.visualstudio.com/items?itemName=fortifyvsts.hpe-security-fortify-vsts



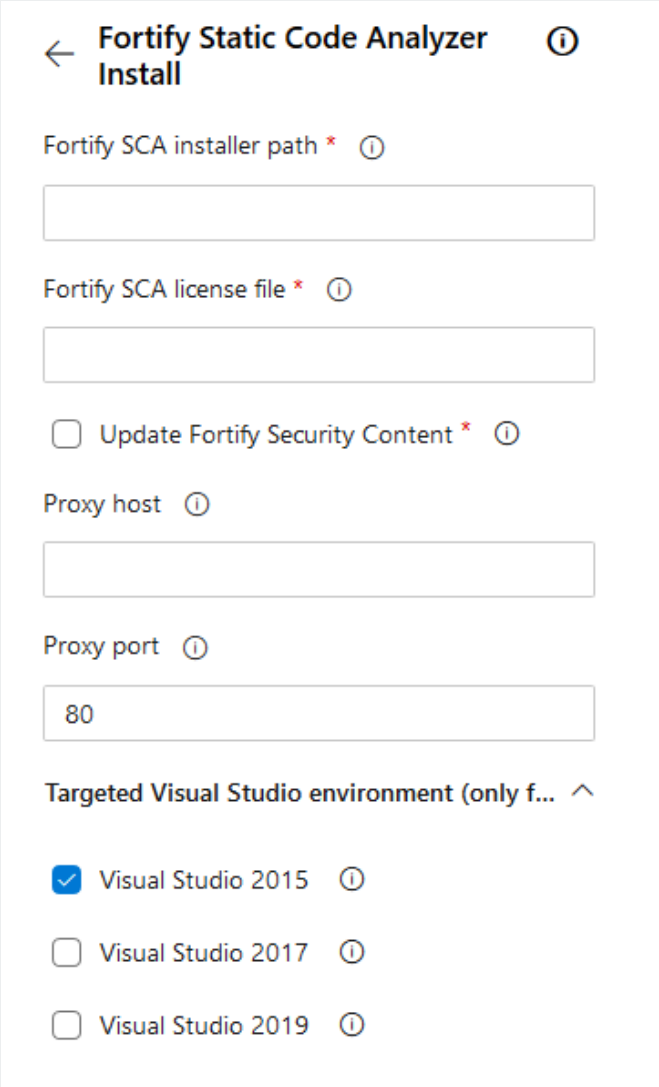
Build Tasks Available

- SAST
  - Fortify Static Code Analyzer Install
  - Fortify Static Code Analyzer Assessment
  - Fortify ScanCentral SAST Assessment
- DAST
  - Fortify WebInspect Dynamic Assessment
  - Fortify ScanCentral DAST Assessment



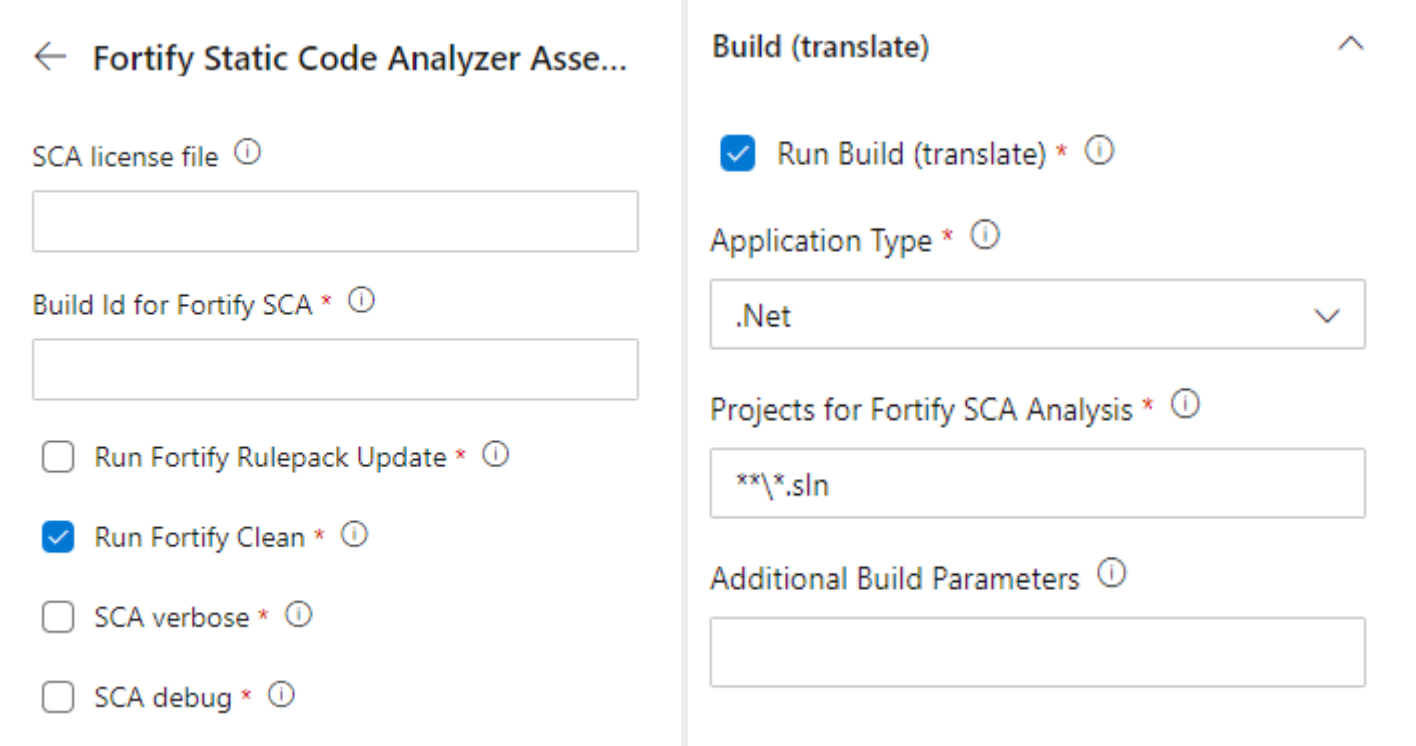
Task - Fortify Static Code Analyzer Install

- Options
  - Select SCA Installer
  - Select Fortify License file
  - Proxy Settings
  - Visual Studio Environment (only for Windows)
- Questions
  - How to copy the installer?
  - After installation sourceanalyzer will be in path?



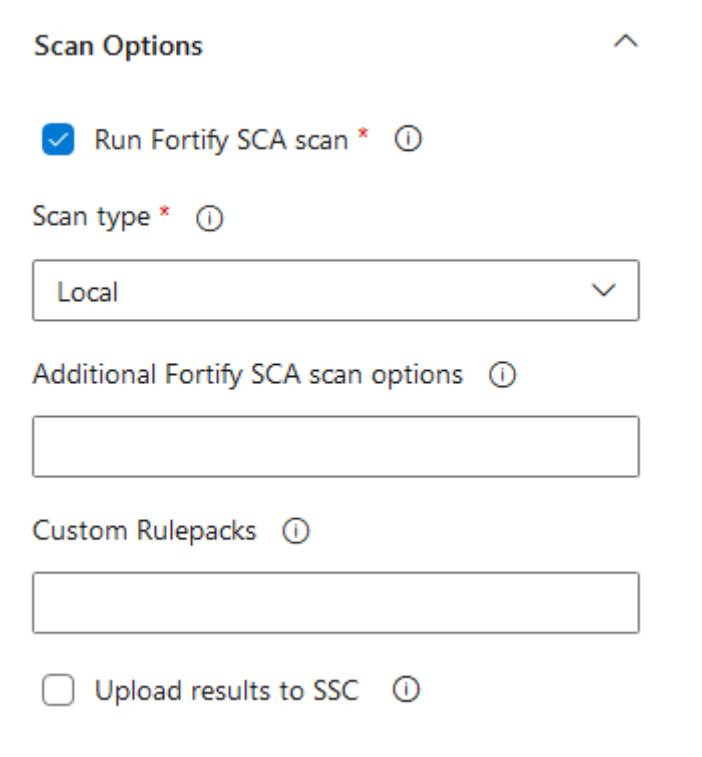
Task - Fortify Static Code Analyzer Assessment

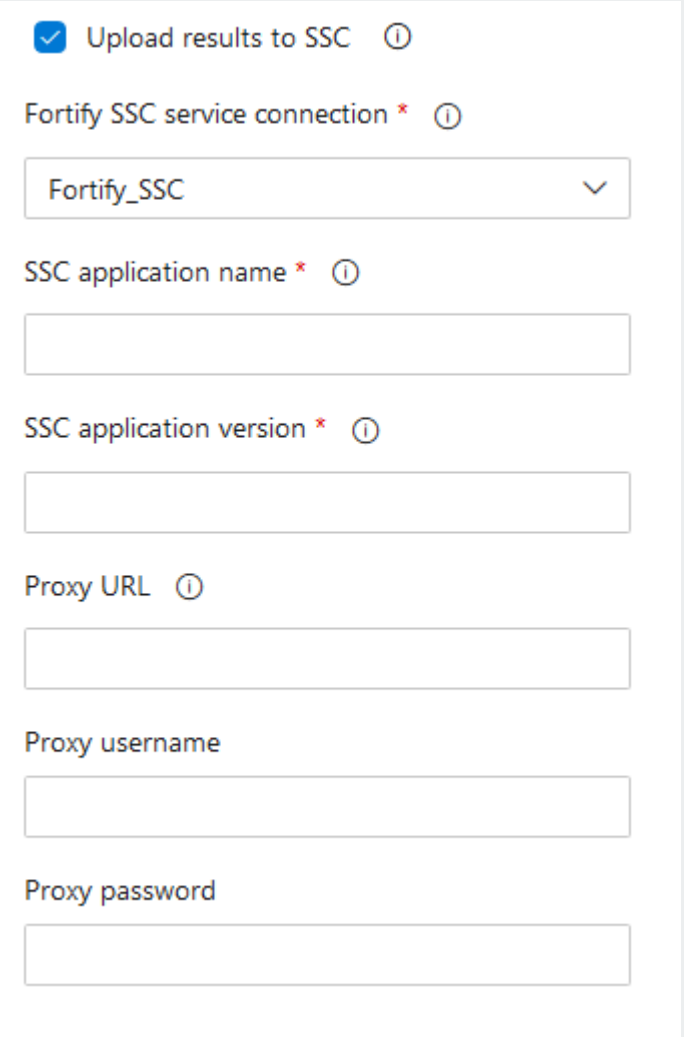
- License File
- Build Id
- Rule Pack Update
- Build Clean
- Build Type
- Projects File



Task - Fortify Static Code Analyzer Assessment - Cont.

- Scan Type
  - Local
  - ScanCentral
- SSC URL
- Application Name
- Application Version
- Proxy details
- No Quality Gate Available – But workaround available





Azure - SCA Integration

DotNet Application - Traditional Approach

```yaml
# Starter pipeline
# Start with a minimal pipeline that you can customize to build and deploy your code.
# Add steps that build, run tests, deploy, and more:
# https://aka.ms/yaml

trigger:
- master

pool:
  vmImage: 'windows-2019'

steps:
Settings
- task: CmdLine@2
  displayName: 'Creating Build Folder for RichesDotnet'
  inputs:
    script: |
      Echo "Creating a target folder"
        MKDIR "D:\hp_la_chouffe\rules\scratch\RichesDotnet\"

```

DotNet Application - Traditional Approach – Cont..

Running MS Build.

```
Settings
- task: MSBuild@1
  inputs:
    solution: '**/*.sln'
    clean: true
    createLogFile: true

```

DotNet Code - Traditional Approach – Cont...

Downloading SCA

Settings

```yaml
- task: PowerShell@2
  inputs:
    targetType: 'inline'
    script: |
      # Write your PowerShell commands here.
      $BaseUrl = "https://[redacted]:8443/ssc/downloads"
      add-type @"
        using System.Net;
        using System.Security.Cryptography.X509Certificates;
        public class TrustAllCertsPolicy : ICertificatePolicy {
          public bool CheckValidationResult(
            ServicePoint srvPoint, X509Certificate certificate,
            WebRequest request, int certificateProblem) {
            return true;
          }
        }
      "@
      [System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy
      Write-Host "Downloading SCA from SSC"
      Invoke-RestMethod http://ipinfo.io/json | Select -exp ip

      New-Item -ItemType directory -Path c:\_agent_distr
      $url = "$BaseUrl/Fortify_SCA_and_Apps_20.2.0_windows_x64.exe"
      $output = "c:\_agent_distr\Fortify_SCA_and_Apps_20.2.0_windows_x64.exe"
      $wc = New-Object System.Net.WebClient
      $wc.DownloadFile($url, $output)

      Write-Host "Downloading fortify.license from SSC"
      $url = "$BaseUrl/fortify.license"
      $output = "c:\_agent_distr\fortify.license"
      Write-Host $url
      $wc = New-Object System.Net.WebClient
      $wc.DownloadFile($url, $output)

      Write-Host "Downloading SSC Cert from SSC"
      $url = "$BaseUrl/SSC.cer"
      $output = "c:\_agent_distr\ssc.cer"
      $wc = New-Object System.Net.WebClient
      $wc.DownloadFile($url, $output)

  displayName: 'Downloading SCA and License'
  enabled: true
```

DotNet Code - Traditional Approach – Cont…

Installing SCA

```yaml
Settings
- task: InstallFortifySCA@7
  inputs:
    InstallerPath: 'c:\_agent_distr\Fortify_SCA_and_Apps_20.2.0_windows_x64.exe'
    VS2015: false
    VS2019: true
    LicenseFile: 'c:\_agent_distr\fortify.license'
    RunFortifyRulepackUpdate: true
  enabled: true
Settings
- task: PowerShell@2
  inputs:
    targetType: 'inline'
    script: |
      Write-Host "Post SCA Install Script"
      Write-Host "##vso[task.prependpath]C:\Fortify\bin\"
      C:\Fortify\jre\bin\keytool.exe -importcert -trustcacerts -alias SSC -keystore C:\Fortify\jre\lib\security\cacerts -file c:\_agent_distr\ssc.cer -storepass changeit -noprompt
  displayName: 'Setting up Path for SCA'
  enabled: true
```

DotNet Code - Traditional Approach – Cont…

Environment Setup

```yaml
- task: BatchScript@1
  inputs:
    filename: 'C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterprise\Common7\Tools\VsDevCmd.bat'
    modifyEnvironment: true
    failOnStandardError: true
  displayName: 'Setting up Visual Studio Environment'
  enabled: true
```

DotNet Code - Traditional Approach – Cont…

Running SCA

```yaml
- task: FortifySCA@6
  inputs:
    licenseFile: 'c:\_agent_distr\fortify.license'
    runBuildTranslate: true
    applicationType: 'dotnet'
    fortifyProjects: '**\*.sln'
    fortifyBuildId: 'RichesDotNet_Azure_DevOps'
    runFortifyRulepackUpdate: true
    runFortifyClean: true
    scaVerbose: true
    scaDebug: true
    runFortifyScan: true
    fortifyScanType: 'LocalScan'
    runFortifyUpload: true
    fortifyServerName: 'Fortify_SSC'
    fortifyApplicationName: 'Riches DotNet Project with Fortify Plugins'
    fortifyApplicationVersion: '1.0'
  enabled: true
```

DotNet Code - Traditional Approach – Cont…

Quality Gate

```
- task: CmdLine@2
  inputs:
    script: |
      Echo Quality Gate for SCA
      FPRUtility -information -search -query "[fortify priority order]:critical" -
project D:\a\1\a\sca_artifacts\RichesDotNet_Azure_DevOps.fpr | findstr /I /C:"No issu
es matched search query."
      rem If %ERRORLEVEL% EQU 0 Echo "Clean Build"
      rem If %ERRORLEVEL% EQU 1 Echo "Dirty Build"
      If %ERRORLEVEL% EQU 0 exit 0
      If %ERRORLEVEL% EQU 1 exit 1
  displayName: 'Running Quality Gate Check'
  enabled: true
```

Azure - ScanCentral Plugin Integration

ScanCentral – Plug-ins Support

Running Maven Build

```yaml
trigger:
- main

pool:
  vmImage: windows-2019

steps:
Settings
- task: Maven@3
  inputs:
    mavenPomFile: 'pom.xml'
    mavenOptions: '-Xmx3072m'
    javaHomeOption: 'JDKVersion'
    jdkVersionOption: '1.8'
    jdkArchitectureOption: 'x64'
    publishJUnitResults: true
    #testResultsFiles: '**/surefire-reports/TEST-*.xml'
    goals: 'package'
  enabled: true
```

ScanCentral – Plug-ins Support

Run the scan using build task plug-ins

Settings

```yaml
- task: FortifyScanCentralSAST@7
  inputs:
    scanCentralCtrlUrl: 'http://[redacted]:8280/scancentral-ctrl'
    scanCentralClientToken: '$(ScanCentral.ClientToken)'
    sscCiToken: '$(ScanCentral.SscCiToken)'
    uploadToSSC: true
    applicationName: 'simplistic.rabbitMQ via ScanCentral'
    applicationVersion: '3.0'
    applicationVersionId: '10005'
    buildTool: 'mvn'
  displayName: 'Fortify ScanCental SAST Scan'
  enabled: true
```

GCP CloudBuild

GCP – Code Build

- Google Cloud – Code Build
- Config File - **cloudbuild.yaml** or cloudbuild.json or docker file
- It runs only docker images
- Supported apps / languages –
    - Node.js
    - Java
    - Python
    - Go
    - VM using Packer

Python Project

cloudbuild.yaml

```yaml
steps:
- name: 'python:3.7.9'
  args: ['python', '-m', 'pip', 'install', '-r', 'requirements.txt', '--user']
- name: 'python:3.7.9'
  args: ['bash', './myscript.bash']
```

myscript.bash for Python Project

```
#!/bin/bash

wget http://[illegible]:8080/ssc/downloads/Fortify_SCA_and_Apps_20.1.0_linux_x64.run

wget http://[illegible]:8080/ssc/downloads/fortify.license

chmod +x ./Fortify_SCA_and_Apps_20.1.0_linux_x64.run

./Fortify_SCA_and_Apps_20.1.0_linux_x64.run --fortify_license_path ./fortify.license
--mode unattended

/opt/Fortify/Fortify_SCA_and_Apps_20.1.0/bin/fortifyupdate

/opt/Fortify/Fortify_SCA_and_Apps_20.1.0/bin/sourceanalyzer -b python_django -debug-
verbose -logfile pyscan.log "./**7*.js" "./**/*.html"

/opt/Fortify/Fortify_SCA_and_Apps_20.1.0/bin/sourceanalyzer -b python_django -debug-
verbose -logfile pyscan.log -python-version 3 -python-path
"/usr/local/lib/python3.7:/usr/local/lib/python3.7/site-
packages:/usr/local/lib64/python3.7:/usr/local/lib64/python3.7/site-
packages:/usr/lib/python3.7:/usr/lib/python3.7/site-
packages:/usr/lib64/python3.7:/usr/lib64/python3.7/site-packages" "./**/*.py"
```

Java Project

cloudbuild.yaml

```yaml
steps:
- name: 'java'
  args: ['bash', './scascan.bash']
  id: 'SCA_Task'
  waitFor: ['-']
  timeout: '1200s'
```

scascan.bash for Java Project

```
#!/bin/bash

wget http://[illegible]:8080/ssc/downloads/Fortify_SCA_and_Apps_20.1.0_linux_x64.run

wget http://[illegible]:8080/ssc/downloads/fortify.license

chmod +x ./Fortify_SCA_and_Apps_20.1.0_linux_x64.run

./Fortify_SCA_and_Apps_20.1.0_linux_x64.run --fortify_license_path ./fortify.license --mode unattended

/opt/Fortify/Fortify_SCA_and_Apps_20.1.0/bin/fortifyupdate

/opt/Fortify/Fortify_SCA_and_Apps_20.1.0/bin/sourceanalyzer -b richsjava -debug-verbose -logfile richsjava.log
"./**/*.js" "./**/*.html" "./**/*.properties" "./**/*.xml" "./**/*.sql" "./**/*.jsp"

/opt/Fortify/Fortify_SCA_and_Apps_20.1.0/bin/sourceanalyzer -b richsjava -debug-verbose -logfile richsjava.log -cp
"jsplibs/**/*.jar:lib/**/*.jar:WEB-INF/lib/**/*.jar" -source 1.5 "./**/*.java"

echo Done with Translate

echo Starting Scan Stage

/opt/Fortify/Fortify_SCA_and_Apps_20.1.0/bin/sourceanalyzer -b richsjava -scan -f richsjava.fpr

echo FPR Generated

/opt/Fortify/Fortify_SCA_and_Apps_20.1.0/bin/fortifyclient -url http://[illegible]:8080/ssc/ -authtoken 70340f5e-
eb79-452f-9a31-c8c475f9a88d -application "Riches Java GCP" -applicationVersion "1.0" uploadFPR -file richsjava.fpr

echo Done with Upload
```

CyberRes

Find. Fix. Fortify.

End-to-end application security.