AgentWorks DevOps, Development, and Tooling

This document was produced as part of offboarding transition of Dmitri Mogilevski at the end of his consulting engagement. This document lists the concerns and responsibilities Dmitri fulfilled that must be transitioned from Dmitri to MoneyGram employees. This document has been used for the employee training and knowledge transfer sessions. It will serve as reference for employees responsible for development and devops on the project.

**Table of contents**

[DevOps concerns 4](#_Toc501356564)

[Development concerns 5](#_Toc501356565)

[Tools List 6](#_Toc501356566)

[DevOps 7](#_Toc501356567)

[Build Server configuration, maintenance, administration, and management 7](#_Toc501356568)

[Information about the server 7](#_Toc501356569)

[Bamboo License 7](#_Toc501356570)

[Bamboo server 7](#_Toc501356571)

[Bamboo service account 8](#_Toc501356572)

[Bamboo server capabilities 8](#_Toc501356573)

[Bamboo User management 8](#_Toc501356574)

[Bamboo Group management 8](#_Toc501356575)

[Bamboo builds configuration, maintenance, administration, and management 8](#_Toc501356576)

[Bamboo Projects 9](#_Toc501356577)

[Bamboo build plan configuration 9](#_Toc501356578)

[Web API build tasks 11](#_Toc501356579)

[TASK — Dump variables to log 11](#_Toc501356580)

[TASK — Source Code Checkout 13](#_Toc501356581)

[TASK — NuGet Package Restore 14](#_Toc501356582)

[TASK — MSBuild 15](#_Toc501356583)

[TASK - Copy Config directory to the Build Drop directory 16](#_Toc501356584)

[TASK — Copy WebConfigs dir to the Build Drop directory 17](#_Toc501356585)

[TASK — Zip deploy package 18](#_Toc501356586)

[TASK — Copy entire build output to the “latest” directory 19](#_Toc501356587)

[TASK — Slack notification 20](#_Toc501356588)

[Static Content build tasks 21](#_Toc501356589)

[TASK — Dump variables to log 21](#_Toc501356590)

[Sample log output 22](#_Toc501356591)

[TASK — Source Code Checkout 24](#_Toc501356592)

[TASK — Script — change directory to AwStaticContent 24](#_Toc501356593)

[TASK — npm — Install global npm dependencies 25](#_Toc501356594)

[TASK — npm — install build dependencies 26](#_Toc501356595)

[TASK — Grunt build 27](#_Toc501356596)

[TASK — Script — Copy build output to the Build Drop directory 28](#_Toc501356597)

[TASK — Script — Zip deploy package 29](#_Toc501356598)

[TASK — Script — Copy build output to the Build Drop latest directory 30](#_Toc501356599)

[TASK — Script — Slack notification 31](#_Toc501356600)

[Builds and Code repositories 32](#_Toc501356601)

[Build to Repository Branch mapping 32](#_Toc501356602)

[Linked Repository naming convention and build plan assignment 33](#_Toc501356603)

[Linked Repositories naming conventions 33](#_Toc501356604)

[Linked Repositories general configurations 33](#_Toc501356605)

[Build Plan configuration — Repositories tab 37](#_Toc501356606)

[Build Plan configuration — Triggers tab 38](#_Toc501356607)

[Build Plan configuration — Permissions tab 38](#_Toc501356608)

[Build Plan configuration — Miscellaneous tab 39](#_Toc501356609)

[Bamboo deploys configuration, maintenance, administration, and management 40](#_Toc501356610)

[Deploy Project and Environment naming conventions 41](#_Toc501356611)

[Deploy Project configuration 42](#_Toc501356612)

[Deploy projects permissions 44](#_Toc501356613)

[Deploy triggers settings 45](#_Toc501356614)

[Deploy Agents assignment 47](#_Toc501356615)

[Deploy variables 47](#_Toc501356616)

[Deploy Environment permissions 48](#_Toc501356617)

[API Deploy tasks 50](#_Toc501356618)

[API Deploy task — Dump variables to log configuration 51](#_Toc501356619)

[Sample variable log dump 52](#_Toc501356620)

[API Deploy task — Clean working directory task configuration 53](#_Toc501356621)

[API Deploy task — Artifact download task configuration 54](#_Toc501356622)

[API Deploy task — Script — XCopy deploy task configuration 55](#_Toc501356623)

[API Deploy task — Script — XCopy deploy task configuration 56](#_Toc501356624)

[API Deploy task — Script — Slack notification task configuration 57](#_Toc501356625)

[Static Content Deploy tasks 58](#_Toc501356626)

[Static Content Deploy task list 58](#_Toc501356627)

[Static Content Deploy task — Dump variables to log task configuration 59](#_Toc501356628)

[Sample log dump 59](#_Toc501356629)

[Static Content Deploy task — Clean working directory task configuration 61](#_Toc501356630)

[Static Content Deploy task — Script — XCopy deploy task configuration 61](#_Toc501356631)

[Static Content Deploy task — Script — Slack notification task configuration 62](#_Toc501356632)

[Bamboo agents 63](#_Toc501356633)

[Build agent configuration 65](#_Toc501356634)

[Adding a new local agent 66](#_Toc501356635)

[Assigning a new local agent to build or deploy 67](#_Toc501356636)

[Bamboo Global variables 68](#_Toc501356637)

[Bamboo administration 68](#_Toc501356638)

[Groups 68](#_Toc501356639)

[Bamboo server backup 69](#_Toc501356640)

[Bamboo server upgrade 71](#_Toc501356641)

[Dev Server configuration, maintenance, administration, and management 72](#_Toc501356642)

[IIS Hosting directory structure 72](#_Toc501356643)

[Dev server logging directory structure 73](#_Toc501356644)

[Dev server environments 73](#_Toc501356645)

[Dev server configuration 74](#_Toc501356646)

[Dev server maintenance 74](#_Toc501356647)

[Git code versioning 75](#_Toc501356648)

[Git install 75](#_Toc501356649)

[Git Client 79](#_Toc501356650)

[Git code versioning, branching and merging 79](#_Toc501356651)

[Versioning 79](#_Toc501356652)

[Branching 80](#_Toc501356653)

[Release candidate branch naming 80](#_Toc501356654)

[Release candidate versioning 80](#_Toc501356655)

[BitBucket repositories management and administration 81](#_Toc501356656)

[BitBucket repository Overview page 82](#_Toc501356657)

[BitBucket repository Commits page 83](#_Toc501356658)

[BitBucket repository Downloads page 84](#_Toc501356659)

[BitBucket repository Settings page 84](#_Toc501356660)

[Development 86](#_Toc501356661)

[Product Owner collaboration 86](#_Toc501356662)

[Story readiness assessment 86](#_Toc501356663)

[Code quality enforcement processes 87](#_Toc501356664)

[Code reviews 87](#_Toc501356665)

[Nuget updates 87](#_Toc501356666)

[NPM Updates 87](#_Toc501356667)

[Authentication and authorization in the Web API 88](#_Toc501356668)

[Summary 88](#_Toc501356669)

[OpenAM/OpenIDM 89](#_Toc501356670)

[User 89](#_Toc501356671)

[Registered device 89](#_Toc501356672)

[Unregistered device 89](#_Toc501356673)

[Fake auth 89](#_Toc501356674)

[API key 89](#_Toc501356675)

[Support key 89](#_Toc501356676)

[Tools List 90](#_Toc501356677)

[Development tools 90](#_Toc501356678)

[Application ecosystem tools 90](#_Toc501356679)

[TransactionRunner 90](#_Toc501356680)

[AcUpgradeTool 91](#_Toc501356681)

[AwLogAnalyzer 91](#_Toc501356682)

[LangProcessor 92](#_Toc501356683)

[Swagger – API documentation and testing 92](#_Toc501356684)

[Trello – knowledge repository 100](#_Toc501356685)

[Add new people to the team 100](#_Toc501356686)

[Slack – team collaboration and DevOps notifications 100](#_Toc501356687)

# DevOps concerns

* Build Server configuration, maintenance, administration, and management
* Bamboo build and deploy configuration, maintenance, administration, and management
* Bamboo problem resolution
* Dev server configuration, maintenance, administration, and management
* Dev server IIS nodes for static content and Web API
* Dev server problem resolution
* Git code versioning, branching and merging
  + Branching strategy
  + Release cycle conventions
* BitBucket repositories management and administration
  + Managing groups and access
  + Branch cleanup
  + Downloads
* Continuous Integration process today
* Continuous Delivery development plan

# Development concerns

* Product Owner collaboration
* Story readiness assessment
* Code quality enforcement processes
* Code reviews
* Nuget updates
* NPM updates
* Architectural integrity maintenance
* External integration patterns (Web API)
  + OpenAM
  + OpenIDM
  + AgentConnect
  + PartnerService
  + DLS
  + Cache
    - AppFabric
    - ActiveSpaces
    - InMemory
    - NoCache
* Caching policy and implementation in Web API
* Caching policy and implementation in Static Content
* Integration tests suite for AC integration
  + Architecture
  + Configuration data
  + Uses
    - Code testing
    - Environment validation
    - Data mapping validation
    - Test data creation
    - TransactionRunner
* Localization and Internationalization in Web API
* Localization and Internationalization in Static Content
* Authentication and authorization in the Web API
  + OpenAM
    - User
    - Registered device
    - Unregistered device
  + Fake auth
  + API key
  + Support key
* Authentication and authorization in the Static Content
  + OpenAM
  + Fake auth
* Fake auth and device registration override for development
* Configuration files in Web API
  + apiKeys.json
  + appSettings.config
  + fakeAuthAgents.json
  + log4net.config
  + supportKeys.json
* Configuration files in Static Content
  + config.json
  + fakeAuthAgents.json

# Tools List

* Development tools
* Application ecosystem tools
  + TransactionRunner
  + AcUpgradeTool
  + AwLogAnalyzer
  + LangProcessor
* Swagger – API documentation and testing
* Trello – knowledge repository
* Slack – team collaboration and DevOps notifications

# DevOps

## Build Server configuration, maintenance, administration, and management

### Information about the server

IP: 172.28.114.109

Host name: dmnacms6863

Memory: 4GB

Disk: C:\ 40GB; D:\ 100GB

OS: Windows Server 2012

Patching Group: Dev\_Corp

To RDP into the server, user needs to be added to the fcbamboobuild FireCall ID and added to CyberArc

UID: AD\fcbamboobuild

PWD: [FireCall pwd] – user network pwd + VIP token

### Bamboo License

<http://172.28.114.109:8085/admin/updateLicense!doDefault.action>

You may view your licensing details or use the update license form to update the license Bamboo is running with.

Organization: MoneyGram International

Date purchased: 19 July 2017

License type: Bamboo Standard: Commercial Server

Number of local agents supported: Unlimited

Number of remote agents supported: 1

Support period lasts until: 19 July 2018

Server id: BTN4-FTPJ-S51D-6FUE

Support entitlement number (sen): SEN-7778725

For updating license, contact Jeff Hoffmann

### Bamboo server

Bamboo server is running on the Build server on port 8085:

<http://172.28.114.109:8085>

Bamboo Version: 6.2.3 build 60210 - 20 Nov 17

Bamboo is running a service: Atlassian Bamboo Bamboo

Install location:

D:\Bamboo

Home location

D:\bamboo-home

### Bamboo service account

UID: [CCDAD0001@ad.moneygram.com](mailto:CCDAD0001@ad.moneygram.com)

PWD: fg5A$D9poft6JM8mi9

This service account is permitted share directory access to target deploy locations, which is how the deploys are set up.

### Bamboo server capabilities

<http://172.28.114.109:8085/admin/agent/configureSharedLocalCapabilities.action>

These are programs Bamboo can use in builds and deploys.

If you need a new capability, install it on the server and then add it through this page.

### Bamboo User management

To add or edit users, navigate to <http://172.28.114.109:8085/admin/user/viewUsers.action>

### Bamboo Group management

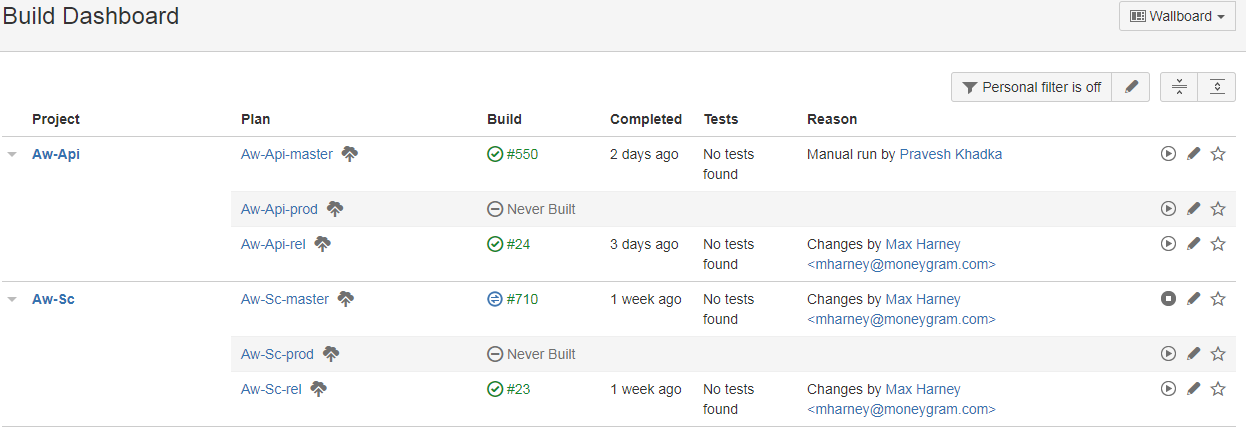
Users are assigned to groups that have permissions over projects, builds and deploys

Groups are managed at <http://172.28.114.109:8085/admin/group/viewGroups.action>

## Bamboo builds configuration, maintenance, administration, and management

Builds projects and plans can be viewed and managed at <http://172.28.114.109:8085/allPlans.action>

Fig. 1 – Bamboo Dashboard

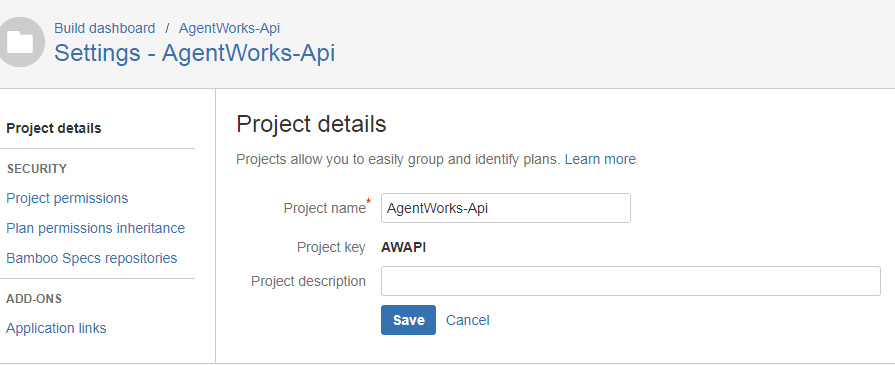


There are 2 projects, one for Web API, one for Static Content

Each project has 3 plans – for master branch, for branch currently in prod, for release candidate and hotfix branch

### Bamboo Projects

Fig. 2 – Bamboo project settings

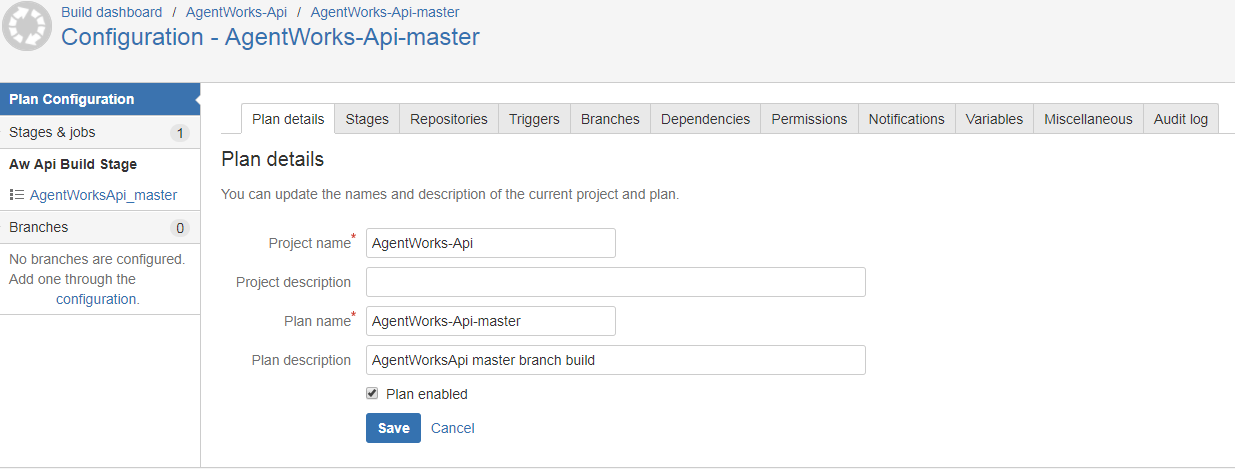


To get to the project settings screen, click on the Project name and then click on the Project Settings button on the right.

### Bamboo build plan configuration

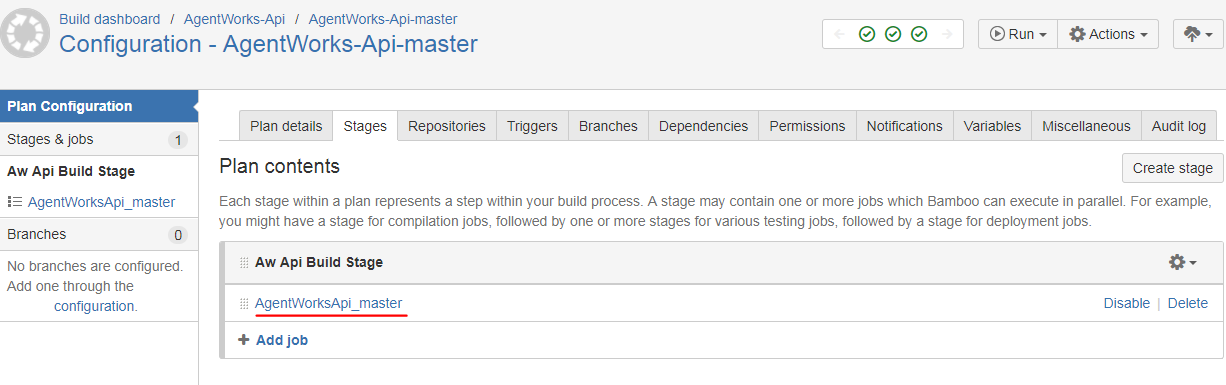
Plans are composed of stages. Our builds have one stage. Once Continuous Delivery processes are implemented (e.g. test environment deploy, smoke test suite execution), additional stages can be defined.

Fig 3. — Plan details tab



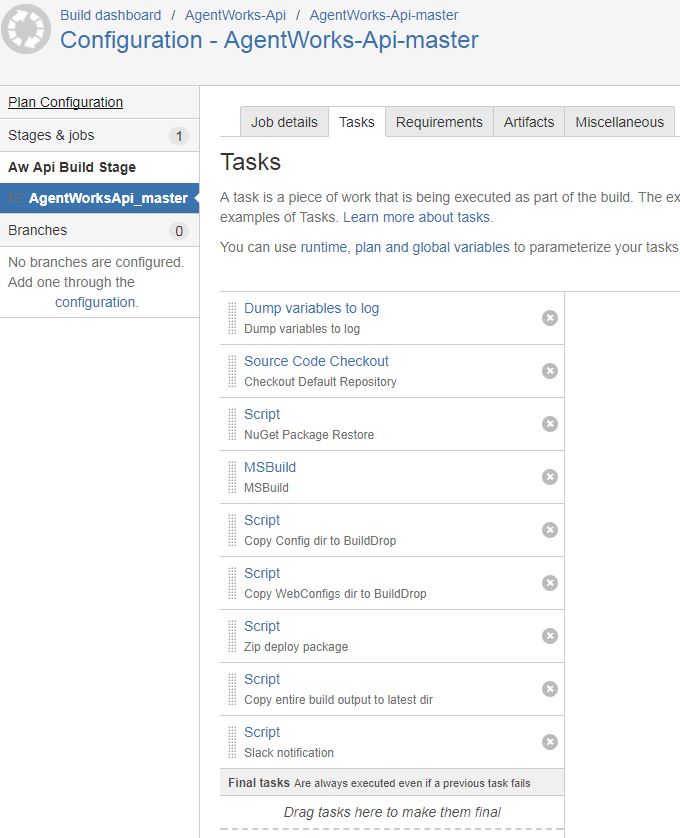
This is where you configure the project name, plan name, and whether the plan is enabled

Fig 4. — Plan stages list view



To configure what the build stage does, click on the stage name.

Fig 5. — Build stage tasks view

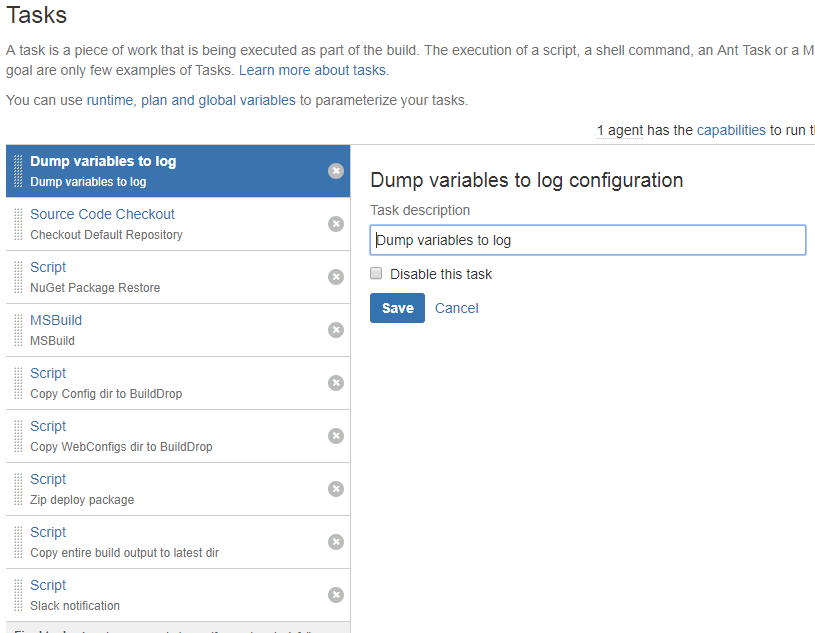


This is the list of tasks, executed consecutively. Clicking on a task allows you to configure or disable it.

### Web API build tasks

#### TASK — Dump variables to log

Fig 5.1 — Dump variables to log task configuration



This task writes all variables used by the build to the build log file. In other tasks, you will see examples of how some of these variables are used. The variables include all accessible variables: global, plan level, Bamboo server level. Here is a sample dump:

simple 21-Nov-2017 14:45:06 ---- DUMPING VARIABLES TO LOG ---

simple 21-Nov-2017 14:45:06 key: [AwApiBuildDrop] value: [D:\AwBuildDrop\AwApi] type: GLOBAL

simple 21-Nov-2017 14:45:06 key: [AwApiD5FtpConnectString] value: [ftp://[UID]:[PWD]@[TARGET IP]] type: GLOBAL

simple 21-Nov-2017 14:45:06 key: [AwApiD5RemoteDir] value: [D:\DeployTarget\D5\AWAPI] type: GLOBAL

simple 21-Nov-2017 14:45:06 key: [AwApiSolutionFileName] value: [AwApi.sln] type: GLOBAL

simple 21-Nov-2017 14:45:06 key: [AwScBuildDrop] value: [D:\AwBuildDrop\AwSc] type: GLOBAL

simple 21-Nov-2017 14:45:06 key: [AwScD5RemoteDir] value: [D:\DeployTarget\D5\AWSC] type: GLOBAL

simple 21-Nov-2017 14:45:06 key: [GlobalNodeModulesDir] value: [C:\Users\fcbamboobuild\AppData\Roaming\npm\node\_modules] type: GLOBAL

simple 21-Nov-2017 14:45:06 key: [NugetExeFullPath] value: ["C:\Program Files (x86)\NuGet\nuget.exe"] type: GLOBAL

simple 21-Nov-2017 14:45:06 key: [agentId] value: [24707076] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [agentWorkingDirectory] value: [D:\bamboo-home\xml-data\build-dir] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [build.working.directory] value: [D:\bamboo-home\xml-data\build-dir\AWAPI-AWAPIM-JOB1] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [buildFailed] value: [false] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [buildKey] value: [AWAPI-AWAPIM-JOB1] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [buildNumber] value: [544] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [buildPlanName] value: [AgentWorks-Api - AgentWorks-Api-master - AgentWorksApi\_master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [buildResultKey] value: [AWAPI-AWAPIM-JOB1-544] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [buildResultsUrl] value: [http://172.28.114.109:8085/browse/AWAPI-AWAPIM-JOB1-544] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [buildTimeStamp] value: [2017-11-21T14:45:06.267-06:00] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.command.MSTest] value: [C:\Program Files (x86)\MSTest\MSTest.exe] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.command.WinSCP.com] value: [C:\Program Files (x86)\winscp577\WinSCP.com] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.command.curl] value: [C:\Program Files\curl] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.command.nuget] value: [C:\Program Files (x86)\NuGet\NuGet.exe] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.devenv.MSTest] value: [C:\Program Files (x86)\MSTest\MSTest.exe] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.msbuild.MSBuild 14] value: [C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.msbuild.MSBuild 15] value: [C:\Program Files (x86)\Microsoft Visual Studio\2017\BuildTools\MSBuild\15.0\Bin\MSBuild.exe] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.msbuild.MSBuild v14.0 (32bit)] value: [C:\Program Files (x86)\MSBuild\14.0\bin\MSBuild.exe] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.msbuild.MSBuild v14.0 (64bit)] value: [C:\Program Files (x86)\MSBuild\14.0\bin\amd64\MSBuild.exe] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.msbuild.MSBuild v4.0 (32bit)] value: [C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.msbuild.MSBuild v4.0 (64bit)] value: [C:\Windows\Microsoft.NET\Framework64\v4.0.30319\MSBuild.exe] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.builder.node.Node.js] value: [C:\Program Files\nodejs\node.exe] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.git.executable] value: [C:\Program Files\Git\cmd\git.exe] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.jdk.JDK] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.jdk.JDK 1.8] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [capability.system.jdk.JDK 1.8.0\_92] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [plan.storageTag] value: [plan-393217] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planKey] value: [AWAPI-AWAPIM] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planName] value: [AgentWorks-Api - AgentWorks-Api-master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.1.branch] value: [master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.1.branchDisplayName] value: [master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.1.branchName] value: [master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.1.name] value: [AgentWorks-Api-master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.1.previousRevision] value: [2229309cf139e09d010afb06143471b97abc11cc] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.1.repositoryUrl] value: [https://bitbucket.org/AgentWorks/agentworksapi] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.1.revision] value: [210b389529bde3d5d69b688a384ad5b3b18af188] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.1.type] value: [bbCloud] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.1.username] value: [ddotm] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.branch] value: [master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.branchDisplayName] value: [master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.branchName] value: [master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.name] value: [AgentWorks-Api-master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.previousRevision] value: [2229309cf139e09d010afb06143471b97abc11cc] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.repositoryUrl] value: [https://bitbucket.org/AgentWorks/agentworksapi] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.revision] value: [210b389529bde3d5d69b688a384ad5b3b18af188] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.type] value: [bbCloud] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [planRepository.username] value: [ddotm] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [repository.12517378.branch.name] value: [master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [repository.12517378.name] value: [AgentWorks-Api-master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [repository.12517378.previous.revision.number] value: [2229309cf139e09d010afb06143471b97abc11cc] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [repository.12517378.revision.number] value: [210b389529bde3d5d69b688a384ad5b3b18af188] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [repository.branch.name] value: [master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [repository.name] value: [AgentWorks-Api-master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [repository.previous.revision.number] value: [2229309cf139e09d010afb06143471b97abc11cc] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [repository.revision.number] value: [210b389529bde3d5d69b688a384ad5b3b18af188] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [resultsUrl] value: [http://172.28.114.109:8085/browse/AWAPI-AWAPIM-JOB1-544] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [shortJobKey] value: [JOB1] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [shortJobName] value: [AgentWorksApi\_master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [shortPlanKey] value: [AWAPIM] type: CUSTOM

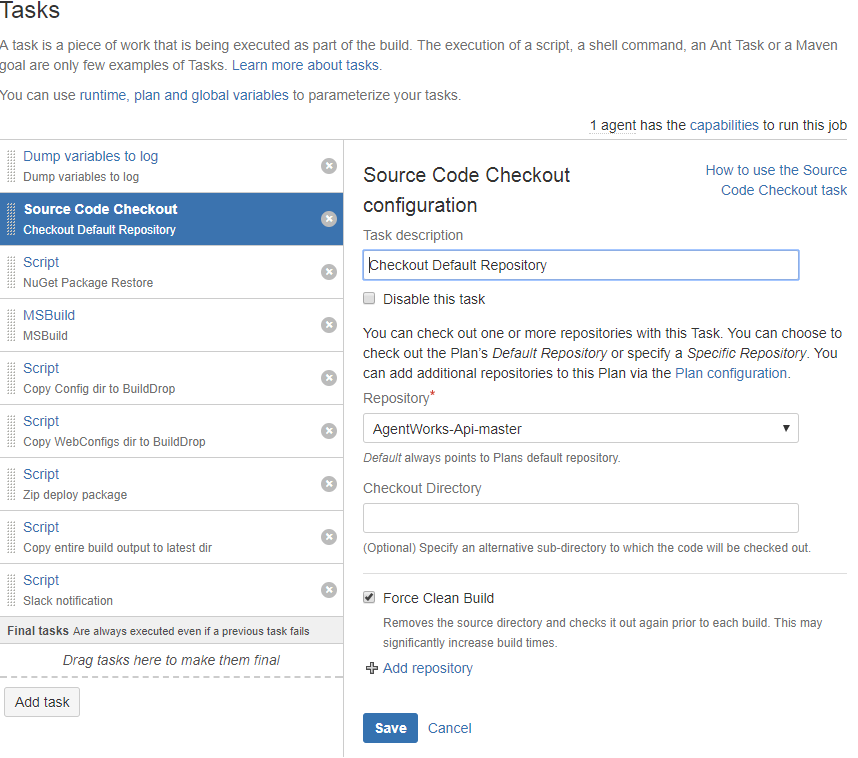
simple 21-Nov-2017 14:45:06 key: [shortPlanName] value: [AgentWorks-Api-master] type: CUSTOM

simple 21-Nov-2017 14:45:06 key: [working.directory] value: [D:\bamboo-home\xml-data\build-dir\AWAPI-AWAPIM-JOB1] type: CUSTOM

simple 21-Nov-2017 14:45:06 ---- END DUMPING VARIABLES TO LOG ---

#### TASK — Source Code Checkout

Fig 5.2 — Source Code Checkout task configuration



The important configuration in this task is the **Repository**.

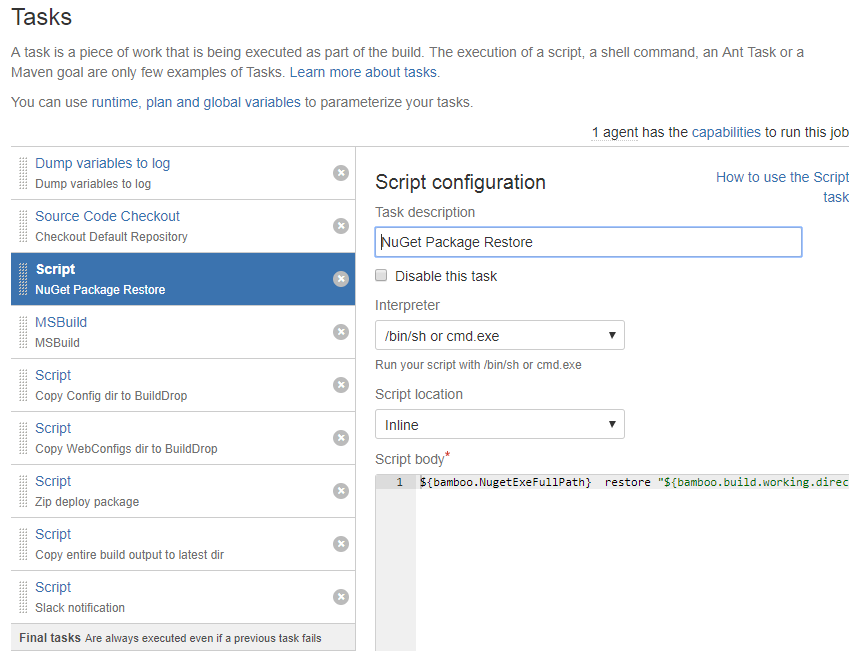
Repositories are configured at <http://172.28.114.109:8085/admin/configureLinkedRepositories!doDefault.action>

For more information on how the linked repositories are configured, read [Linked Repository naming convention and build plan assignment](#_Linked_Repository_naming)

Make sure “Force Clean Build” is checked. This ensures that the entire source is pulled and any structural changes (new projects, deleted projects, solution updates) are pulled.

#### TASK — NuGet Package Restore

Fig 5.3 — Script — NuGet Package Restore task configuration



Script body

${bamboo.NugetExeFullPath} restore "${bamboo.build.working.directory}\${bamboo.AwApiSolutionFileName}"

This is an example of how Bamboo variables are used. The format is ${bamboo.VARIABLE\_NAME}. Bamboo simply interpolates the string that is assigned to the variable into the result string.

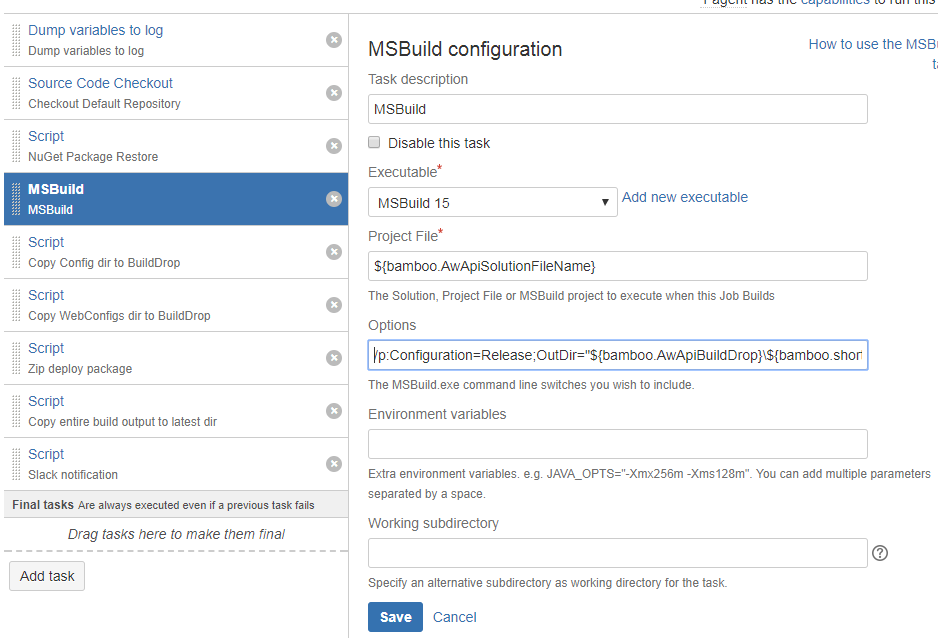
The script command above executes as

"C:\Program Files (x86)\NuGet\nuget.exe" restore "D:\bamboo-home\xml-data\build-dir\AWAPI-AWAPIM-JOB1\AwApi.sln"

This task downloads all nuget package binaries.

#### TASK — MSBuild

Fig 5.4 — MSBuild task configuration



The executable must be set to the version of MSBuild that matches the Visual Studio version. For 2017, MSBuild version is 15.

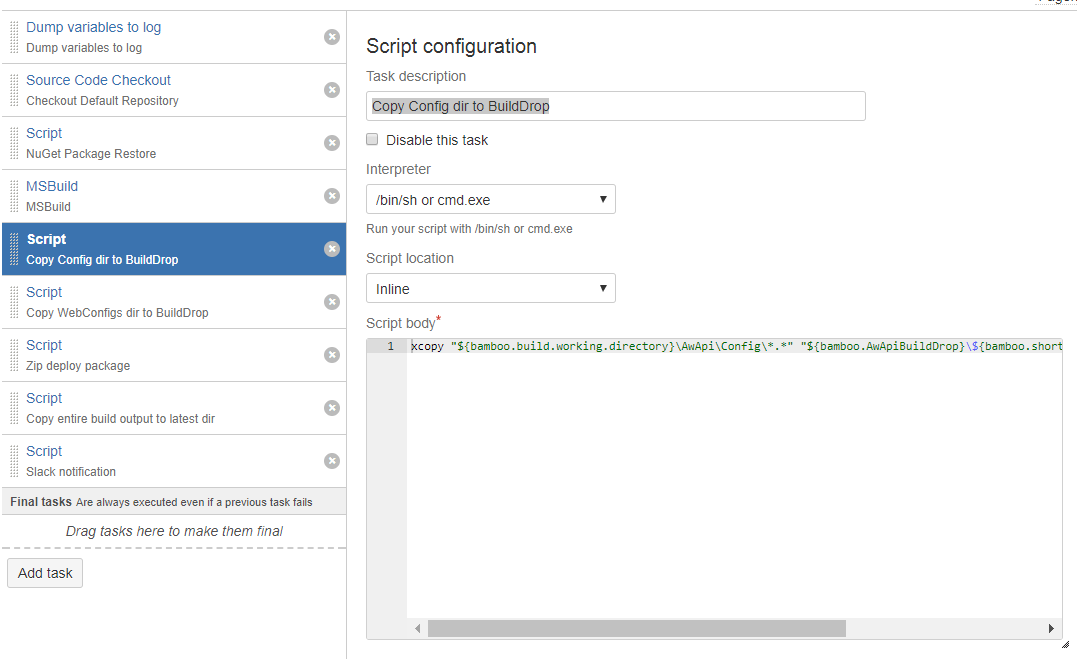
Project File points to the solution file path

“Options” configures which build configuration in the solution gets executed.

After this task, the application is built. Subsequent steps package build artifacts for deployment.

#### TASK - Copy Config directory to the Build Drop directory

Fig 5.5 — Script — Copy Config dir to BuildDrop configuration



Script body

xcopy "${bamboo.build.working.directory}\AwApi\Config\\*.\*" "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\Config\" /E /Y

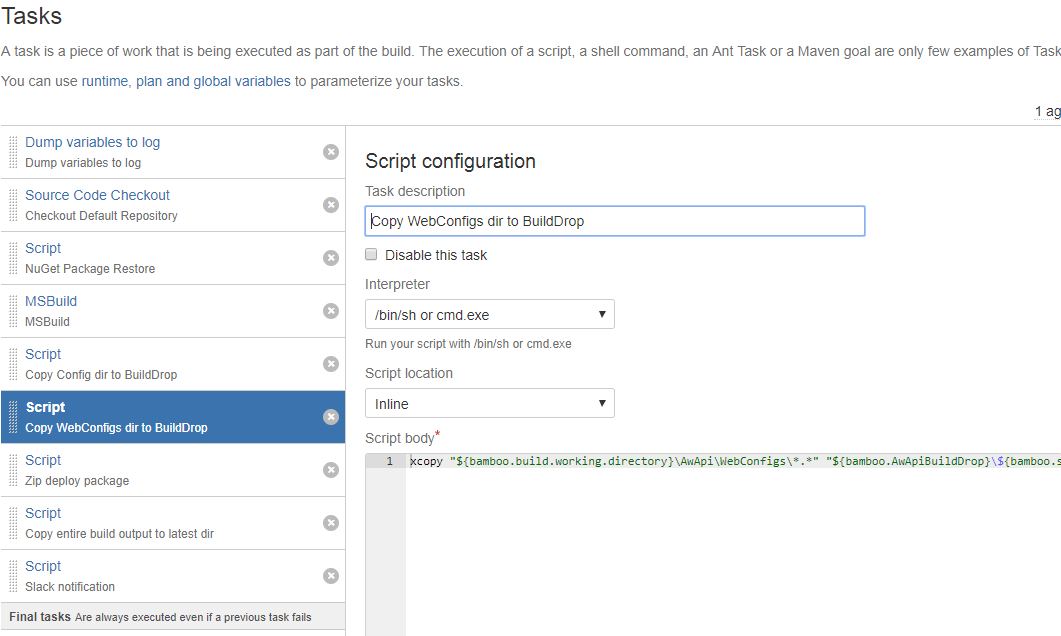
/E /Y flags are used because:

/e Copies all subdirectories, even if they are empty

/y Suppresses prompting to confirm that you want to overwrite an existing destination file.

#### TASK — Copy WebConfigs dir to the Build Drop directory

Fig 5.6 — Script — Copy WebConfigs directory to BuildDrop

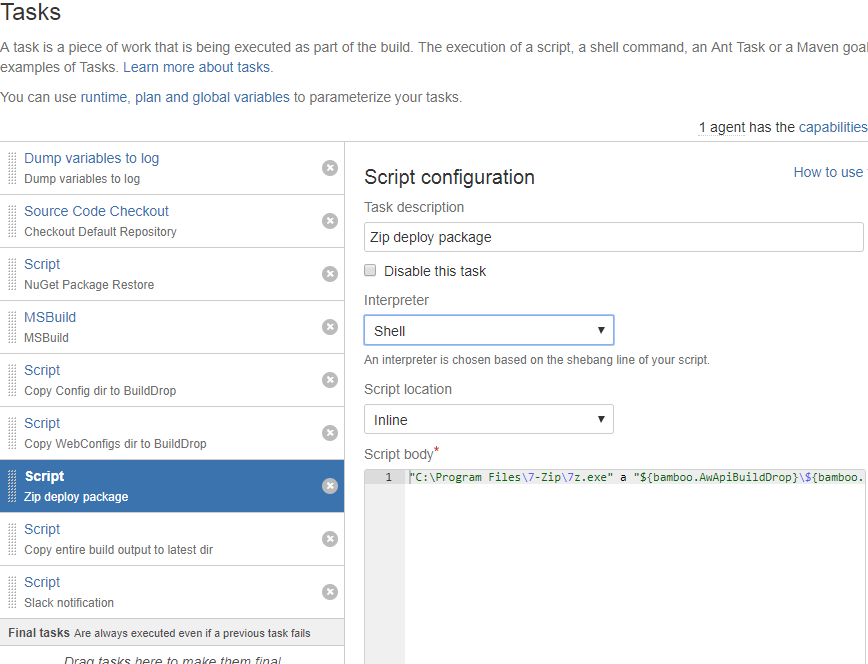


Script body

xcopy "${bamboo.build.working.directory}\AwApi\WebConfigs\\*.\*" "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\WebConfigs\" /E /Y

#### TASK — Zip deploy package

Fig 5.7 — Script — Zip deploy package configuration



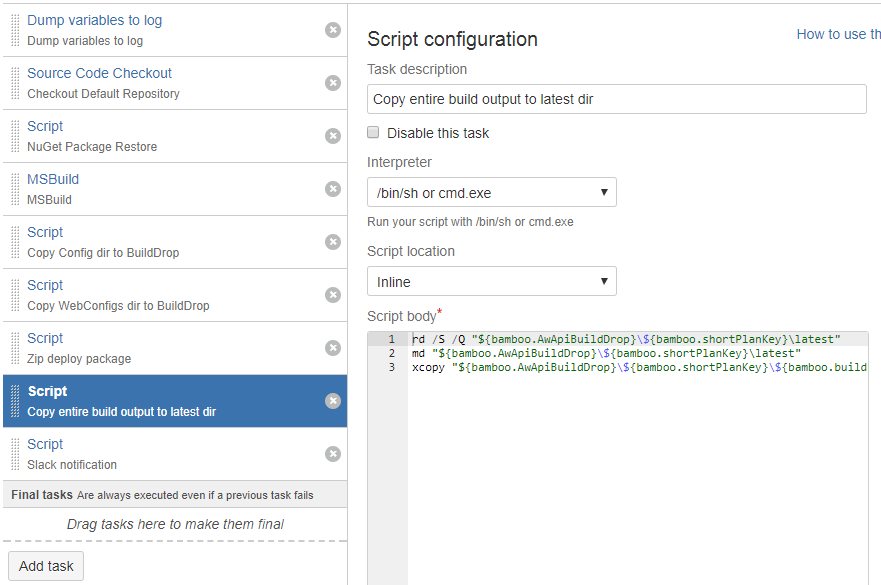
Interpreter is set to Shell

Script body

"C:\Program Files\7-Zip\7z.exe" a "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\${bamboo.shortPlanKey}\_${bamboo.buildNumber}.zip" "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}"

#### TASK — Copy entire build output to the “latest” directory

Fig 5.8 — Script — Copy entire build output to latest dir configuration



Script body

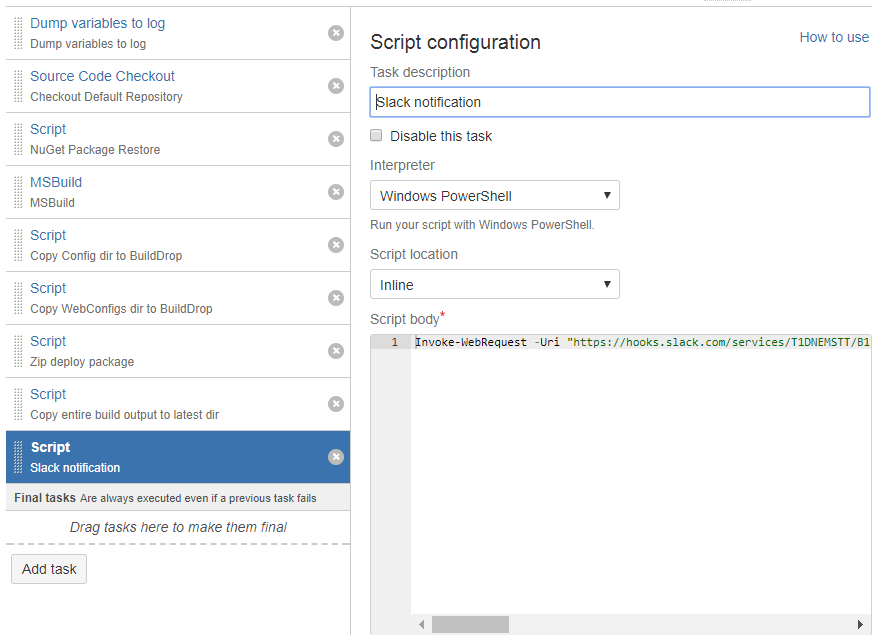
rd /S /Q "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\latest"

md "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\latest"

xcopy "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\\*.\*" "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\latest\" /E /Y

#### TASK — Slack notification

Fig 5.9 — Script — Slack notification configuration



Interpreter: Windows PowerShell

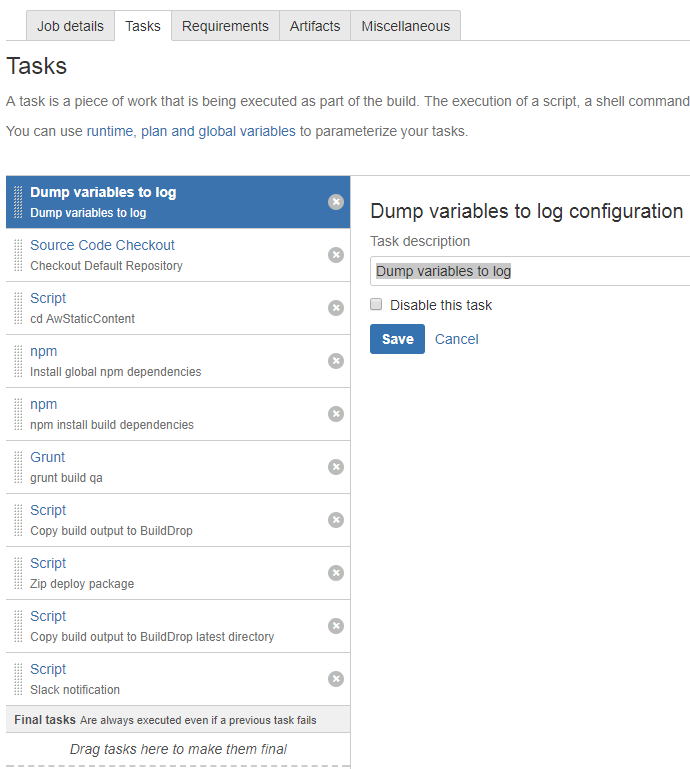
Script body

Invoke-WebRequest -Uri "https://hooks.slack.com/services/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" -Body "{""text"": ""${bamboo.planRepository.name}: \*BUILD\* ${bamboo.buildResultKey} for branch ${bamboo.planRepository.branchName} was \*SUCCESSFUL\*. View results at <${bamboo.resultsUrl}|${bamboo.resultsUrl}>""}" -ContentType "application/json" -Method Post -UseBasicParsing

### Static Content build tasks

#### TASK — Dump variables to log

Fig 5.10 — Dump variables to log configuration



#### Sample log output

simple 22-Nov-2017 10:06:51 ---- DUMPING VARIABLES TO LOG ---

simple 22-Nov-2017 10:06:51 key: [AwApiBuildDrop] value: [D:\AwBuildDrop\AwApi] type: GLOBAL

simple 22-Nov-2017 10:06:51 key: [AwApiD5FtpConnectString] value: [ftp://[UID]:[PWD]@[TARGET IP]] type: GLOBAL

simple 22-Nov-2017 10:06:51 key: [AwApiD5RemoteDir] value: [D:\DeployTarget\D5\AWAPI] type: GLOBAL

simple 22-Nov-2017 10:06:51 key: [AwApiSolutionFileName] value: [AwApi.sln] type: GLOBAL

simple 22-Nov-2017 10:06:51 key: [AwScBuildDrop] value: [D:\AwBuildDrop\AwSc] type: GLOBAL

simple 22-Nov-2017 10:06:51 key: [AwScD5RemoteDir] value: [D:\DeployTarget\D5\AWSC] type: GLOBAL

simple 22-Nov-2017 10:06:51 key: [GlobalNodeModulesDir] value: [C:\Users\fcbamboobuild\AppData\Roaming\npm\node\_modules] type: GLOBAL

simple 22-Nov-2017 10:06:51 key: [NugetExeFullPath] value: ["C:\Program Files (x86)\NuGet\nuget.exe"] type: GLOBAL

simple 22-Nov-2017 10:06:51 key: [agentId] value: [24707073] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [agentWorkingDirectory] value: [D:\bamboo-home\xml-data\build-dir] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [build.working.directory] value: [D:\bamboo-home\xml-data\build-dir\AWSC-AWSCM-JOB1] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [buildFailed] value: [false] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [buildKey] value: [AWSC-AWSCM-JOB1] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [buildNumber] value: [709] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [buildPlanName] value: [AgentWorks-StaticContent - AgentWorks-Sc-master - Aw static content build] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [buildResultKey] value: [AWSC-AWSCM-JOB1-709] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [buildResultsUrl] value: [http://172.28.114.109:8085/browse/AWSC-AWSCM-JOB1-709] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [buildTimeStamp] value: [2017-11-22T10:06:51.266-06:00] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.command.MSTest] value: [C:\Program Files (x86)\MSTest\MSTest.exe] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.command.WinSCP.com] value: [C:\Program Files (x86)\winscp577\WinSCP.com] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.command.curl] value: [C:\Program Files\curl] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.command.nuget] value: [C:\Program Files (x86)\NuGet\NuGet.exe] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.devenv.MSTest] value: [C:\Program Files (x86)\MSTest\MSTest.exe] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.msbuild.MSBuild 14] value: [C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.msbuild.MSBuild 15] value: [C:\Program Files (x86)\Microsoft Visual Studio\2017\BuildTools\MSBuild\15.0\Bin\MSBuild.exe] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.msbuild.MSBuild v14.0 (32bit)] value: [C:\Program Files (x86)\MSBuild\14.0\bin\MSBuild.exe] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.msbuild.MSBuild v14.0 (64bit)] value: [C:\Program Files (x86)\MSBuild\14.0\bin\amd64\MSBuild.exe] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.msbuild.MSBuild v4.0 (32bit)] value: [C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.msbuild.MSBuild v4.0 (64bit)] value: [C:\Windows\Microsoft.NET\Framework64\v4.0.30319\MSBuild.exe] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.builder.node.Node.js] value: [C:\Program Files\nodejs\node.exe] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.git.executable] value: [C:\Program Files\Git\cmd\git.exe] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.jdk.JDK] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.jdk.JDK 1.8] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [capability.system.jdk.JDK 1.8.0\_92] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [plan.storageTag] value: [plan-1343489] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planKey] value: [AWSC-AWSCM] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planName] value: [AgentWorks-StaticContent - AgentWorks-Sc-master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.1.branch] value: [master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.1.branchDisplayName] value: [master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.1.branchName] value: [master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.1.name] value: [AgentWorks-Sc-master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.1.previousRevision] value: [fd56d1eacb65fef7fcf4165147603a0ad104be4d] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.1.repositoryUrl] value: [https://bitbucket.org/AgentWorks/agentworksstaticcontent] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.1.revision] value: [9e0ee3d23adbd409ef373d822d3fa71b21414dd4] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.1.type] value: [bbCloud] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.1.username] value: [ddotm] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.branch] value: [master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.branchDisplayName] value: [master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.branchName] value: [master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.name] value: [AgentWorks-Sc-master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.previousRevision] value: [fd56d1eacb65fef7fcf4165147603a0ad104be4d] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.repositoryUrl] value: [https://bitbucket.org/AgentWorks/agentworksstaticcontent] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.revision] value: [9e0ee3d23adbd409ef373d822d3fa71b21414dd4] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.type] value: [bbCloud] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [planRepository.username] value: [ddotm] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [repository.12517380.branch.name] value: [master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [repository.12517380.name] value: [AgentWorks-Sc-master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [repository.12517380.previous.revision.number] value: [fd56d1eacb65fef7fcf4165147603a0ad104be4d] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [repository.12517380.revision.number] value: [9e0ee3d23adbd409ef373d822d3fa71b21414dd4] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [repository.branch.name] value: [master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [repository.name] value: [AgentWorks-Sc-master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [repository.previous.revision.number] value: [fd56d1eacb65fef7fcf4165147603a0ad104be4d] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [repository.revision.number] value: [9e0ee3d23adbd409ef373d822d3fa71b21414dd4] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [resultsUrl] value: [http://172.28.114.109:8085/browse/AWSC-AWSCM-JOB1-709] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [shortJobKey] value: [JOB1] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [shortJobName] value: [Aw static content build] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [shortPlanKey] value: [AWSCM] type: CUSTOM

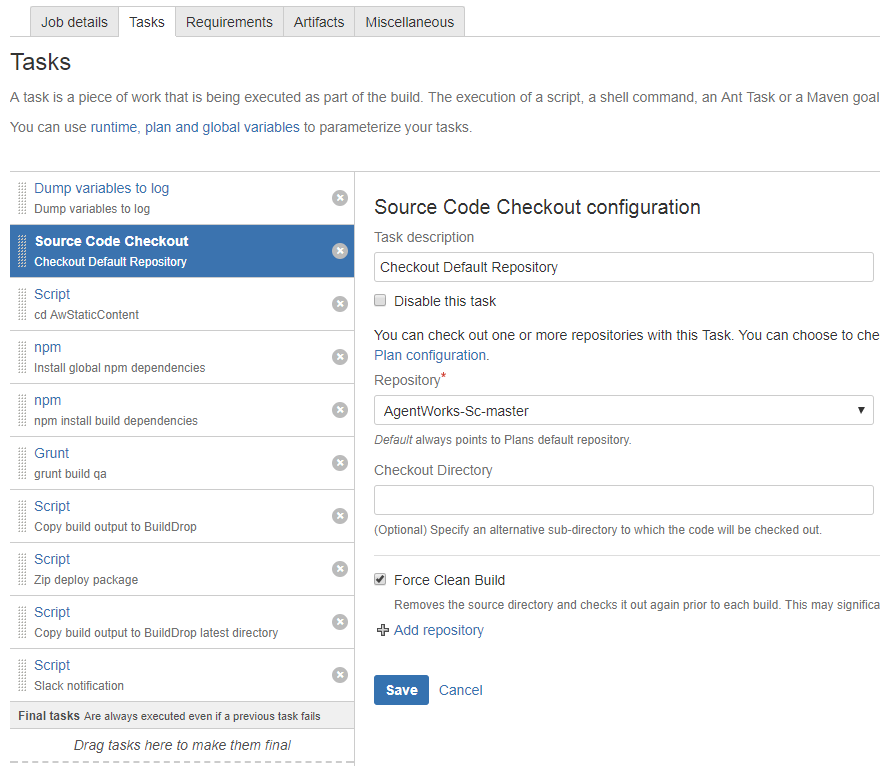
simple 22-Nov-2017 10:06:51 key: [shortPlanName] value: [AgentWorks-Sc-master] type: CUSTOM

simple 22-Nov-2017 10:06:51 key: [working.directory] value: [D:\bamboo-home\xml-data\build-dir\AWSC-AWSCM-JOB1] type: CUSTOM

simple 22-Nov-2017 10:06:51 ---- END DUMPING VARIABLES TO LOG ---

#### TASK — Source Code Checkout

Fig 5.11 — Source Code Checkout configuration



The important configuration in this task is the **Repository**.

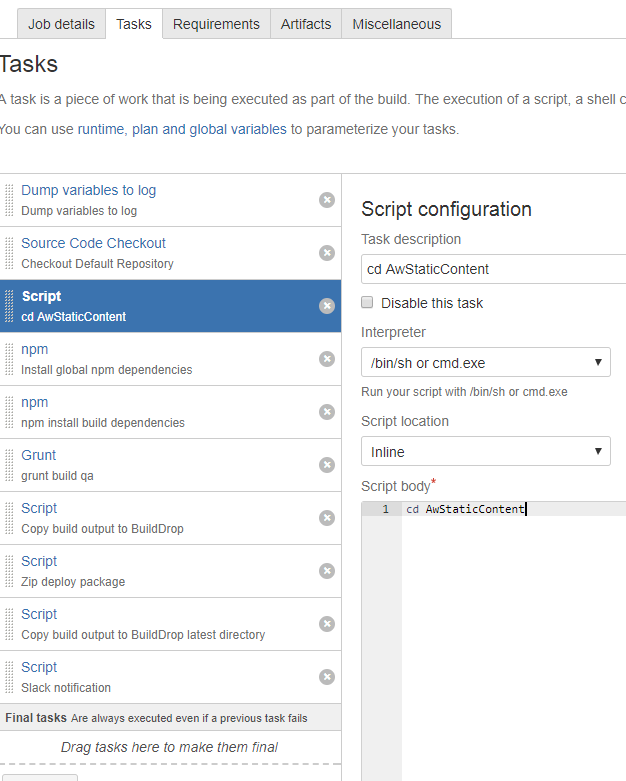
Repositories are configured at <http://172.28.114.109:8085/admin/configureLinkedRepositories!doDefault.action>

For more information on how the linked repositories are configured, read [Linked Repository naming convention and build plan assignment](#_Linked_Repository_naming)

Make sure “Force Clean Build” is checked. This ensures that the entire source is pulled and any structural changes (new projects, deleted projects, solution updates) are pulled.

#### TASK — Script — change directory to AwStaticContent

Fig 5.12 — cd AwStaticContent configuration

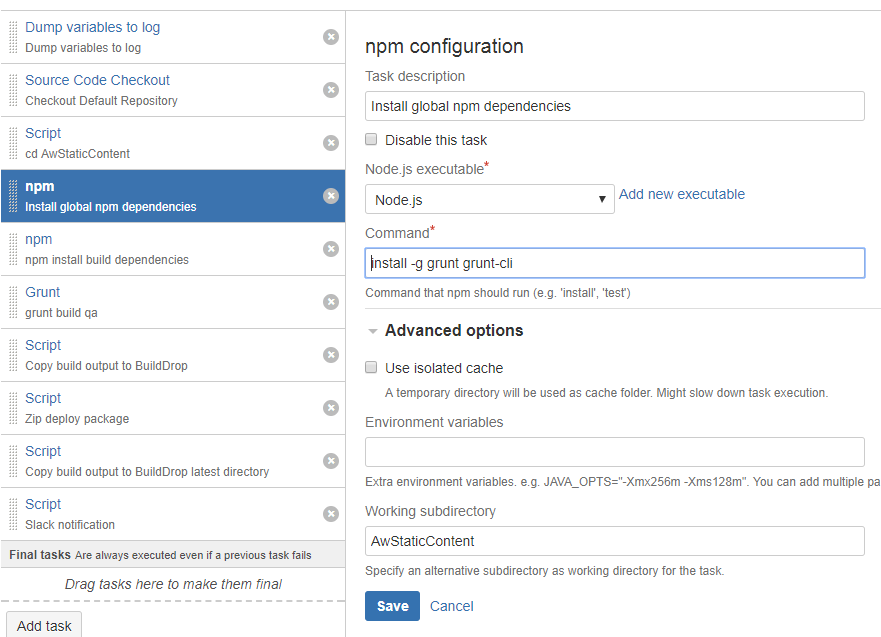


cd AwStaticContent

This moved the executing context to AwStaticContent directory where all configuration and source code is.

#### TASK — npm — Install global npm dependencies

Fig 5.13 — Install global npm dependencies configuration



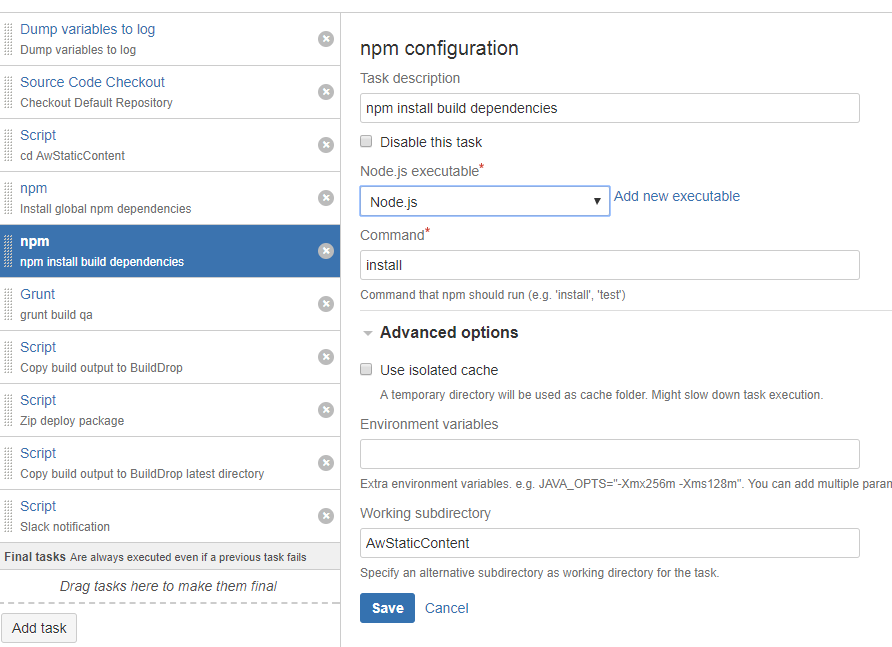
Command:

install -g grunt grunt-cli

This is the task that installs npm packages that the build needs to be globally accessible.

#### TASK — npm — install build dependencies

Fig 5.14 — npm install build dependencies configuration



This task installs dependencies (JavaScript and CSS libraries used by the app) and devDependencies (npm packages used by the build) listed in package.json.

#### TASK — Grunt build

Fig 5.14 — Grunt build configuration



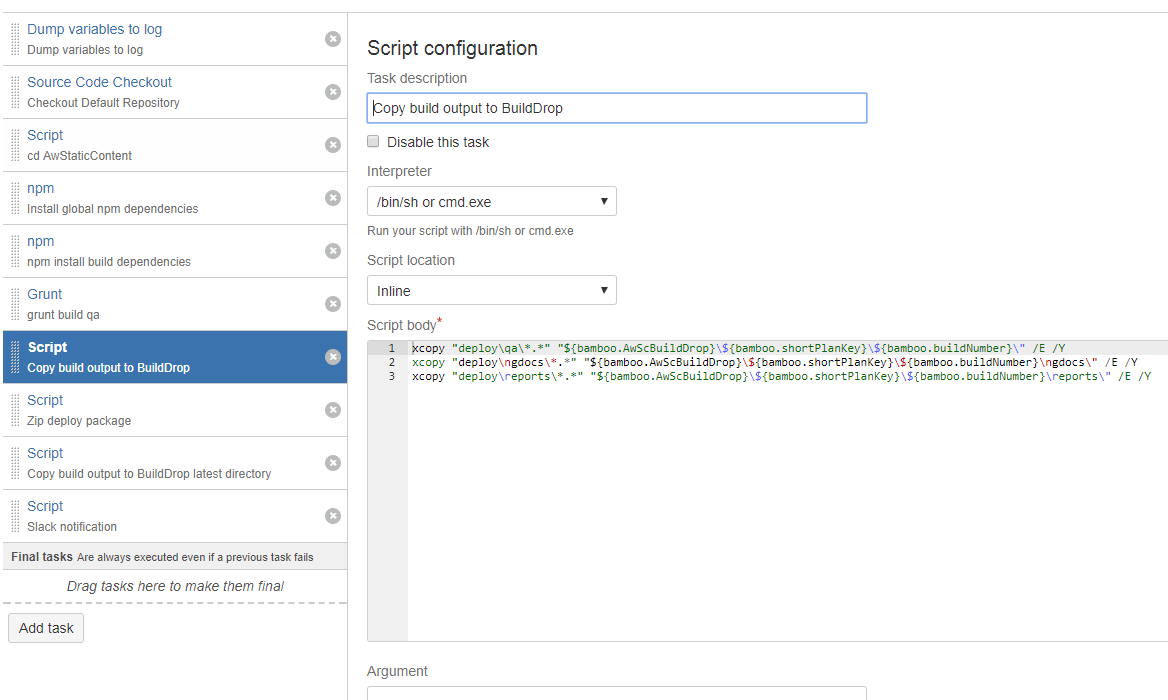
“Task” is set to the grunt build configuration that will be executed.

qa build contains all tasks for the production build. The main differences with the dev build are source minification, uglification, concatenation into one js and one css file, and documentation tasks.

At this point, the build artifacts are completely done. The rest of the build packages the artifacts for deployment.

#### TASK — Script — Copy build output to the Build Drop directory

Fig 5.15 — Copy build output to BuildDrop configuration



Script body

xcopy "deploy\qa\\*.\*" "${bamboo.AwScBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\" /E /Y

xcopy "deploy\ngdocs\\*.\*" "${bamboo.AwScBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\ngdocs\" /E /Y

xcopy "deploy\reports\\*.\*" "${bamboo.AwScBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\reports\" /E /Y

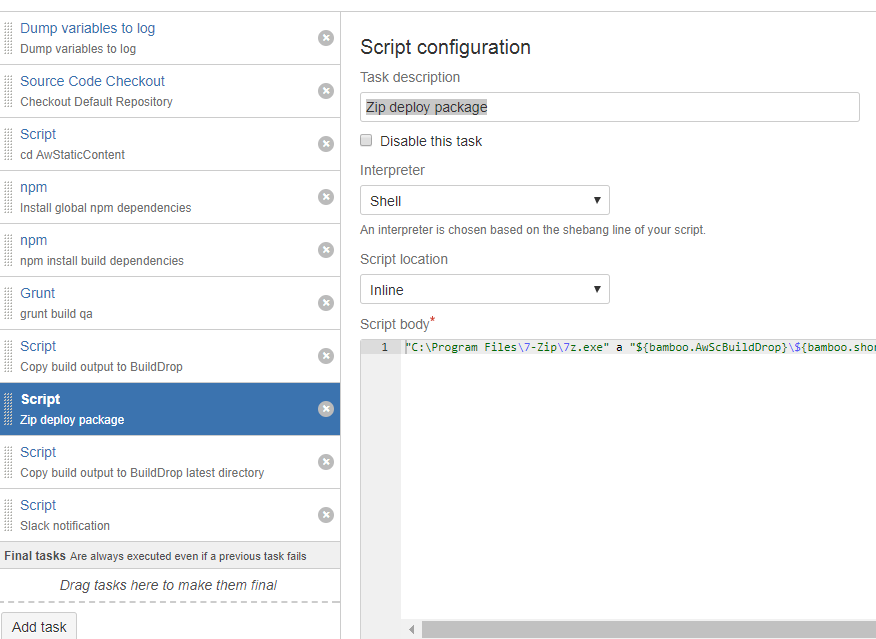
The first command copies the application code artifacts. This is the actual AW static content app.

The second command copies the application documentation site. This is generated from the ng-doc markup in the source code. For more information on ng-doc syntax usage for AngularJs, refer to <https://github.com/angular/angular.js/wiki/Writing-AngularJS-Documentation>

The third command copies the unit test coverage site.

#### TASK — Script — Zip deploy package

Fig 5.16 — Zip deploy package configuration

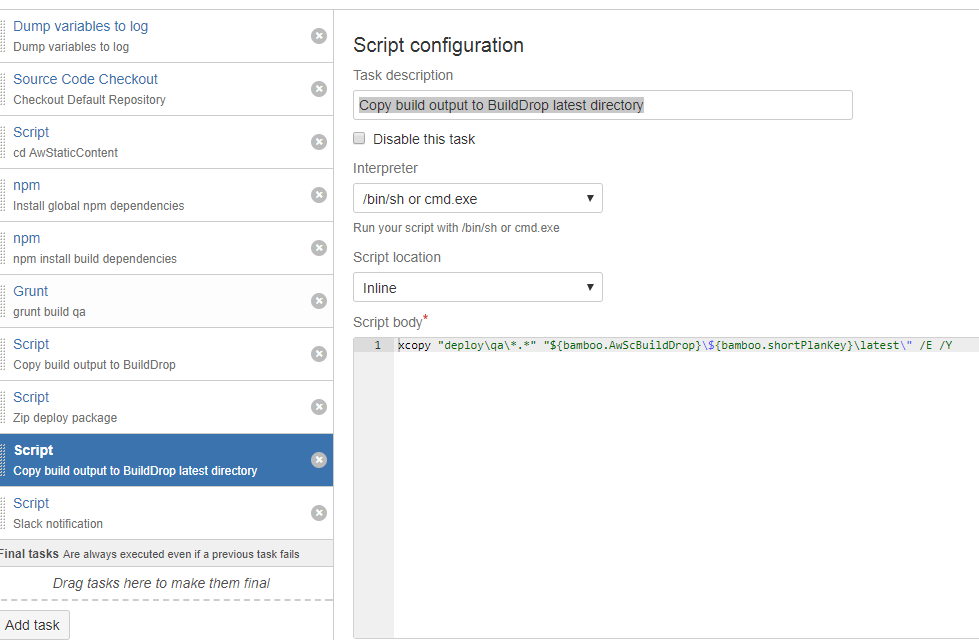


Script body

${bamboo.7ZipExeFullPath} a "${bamboo.AwScBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\${bamboo.shortPlanKey}\_${bamboo.buildNumber}.zip" "${bamboo.AwScBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}"

#### TASK — Script — Copy build output to the Build Drop latest directory

Fig 5.17 — Copy to BuildDrop latest dir configuration

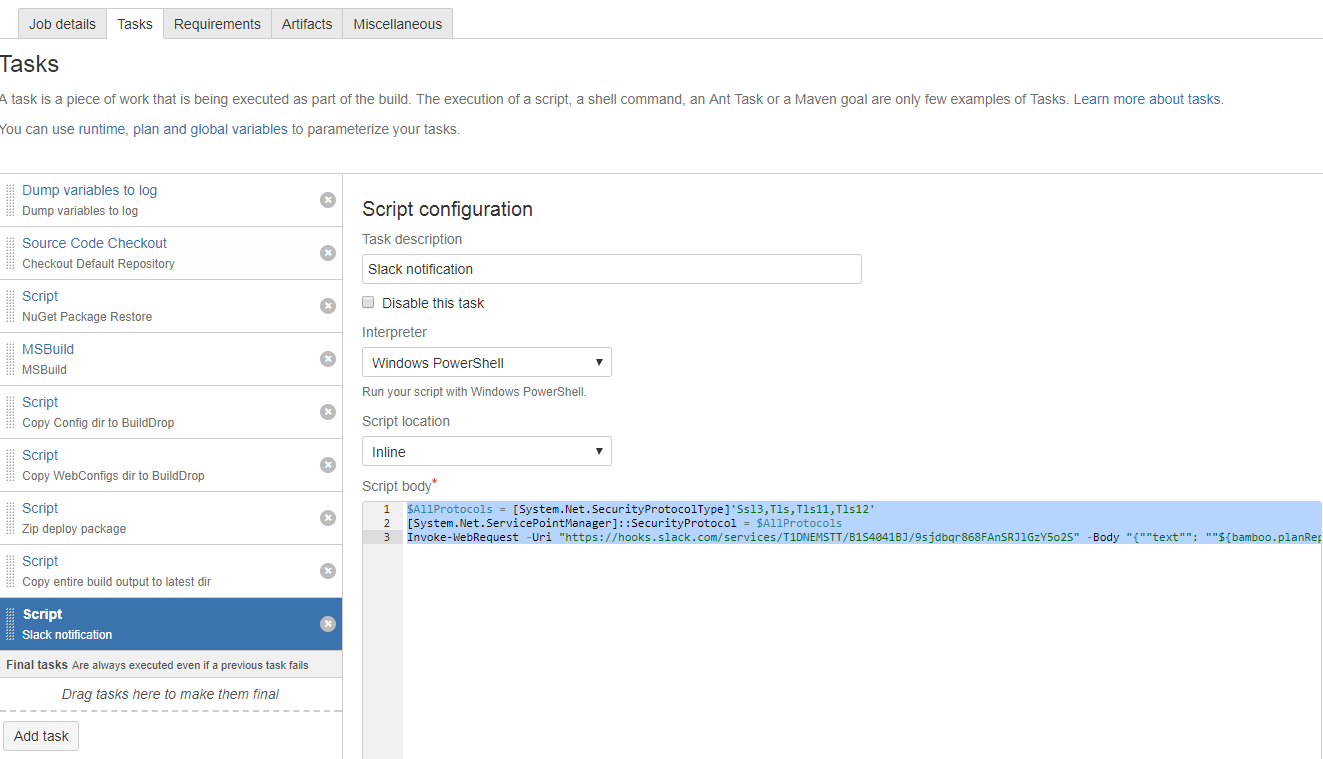


Script body

xcopy "deploy\qa\\*.\*" "${bamboo.AwScBuildDrop}\${bamboo.shortPlanKey}\latest\" /E /Y

#### TASK — Script — Slack notification

Fig 5.18 — Slack notification configuration



Interpreter: Windows PowerShell

Script body

$AllProtocols = [System.Net.SecurityProtocolType]'Ssl3,Tls,Tls11,Tls12'

[System.Net.ServicePointManager]::SecurityProtocol = $AllProtocols

Invoke-WebRequest -Uri "https://hooks.slack.com/services/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" -Body "{""text"": ""${bamboo.planRepository.name}: \*BUILD\* ${bamboo.buildResultKey} for branch ${bamboo.planRepository.branchName} was \*SUCCESSFUL\*. View results at <${bamboo.resultsUrl}|${bamboo.resultsUrl}>""}" -ContentType "application/json" -Method Post -UseBasicParsing

### Builds and Code repositories

Builds are configured to be triggered by code commits. There are currently 3 build plans defined for each app (Web API and Static Content).

#### Build to Repository Branch mapping

|  |  |
| --- | --- |
| **Build plan name** | **Branch** |
| Aw-Api-master  Aw-Sc-master | Branch intended for the next production release prior to cutting a release candidate branch |
| [Aw-Api-prod](http://172.28.114.109:8085/browse/AWAPI-AWAPIDT4MO)  Aw-Sc-prod | Branch that is currently deployed to production |
| Aw-Api-rel  Aw-Sc-rel | Branch that is in pre-release versioning stages — alpha, beta, release candidate |

This should cover all common development scenarios. If there is a need for a long-standing feature branch to be isolated from the next-to-production branch, another set of

builds should be created to point to that specific feature branch. One example of this was when the team worked on AC16.11 integration in parallel to money orders. Money

Orders work was done in the master branch, while AC16.11 was done in a separate dedicated AC16.11 branch.

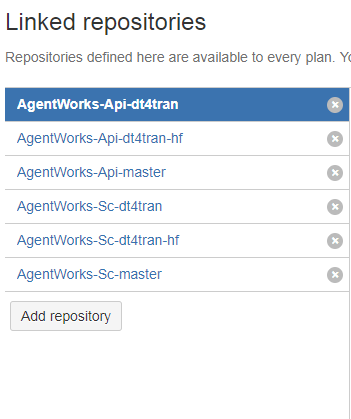
#### Linked Repository naming convention and build plan assignment

While the Build plan name and its relationship to the release cycle never changes, the linked repositories will be updated as the application goes through releases.

The place to define and manage linked repositories in Bamboo is <http://172.28.114.109:8085/admin/configureLinkedRepositories!doDefault.action>

##### Linked Repositories naming conventions

Fig. 6 – Linked Repositories list



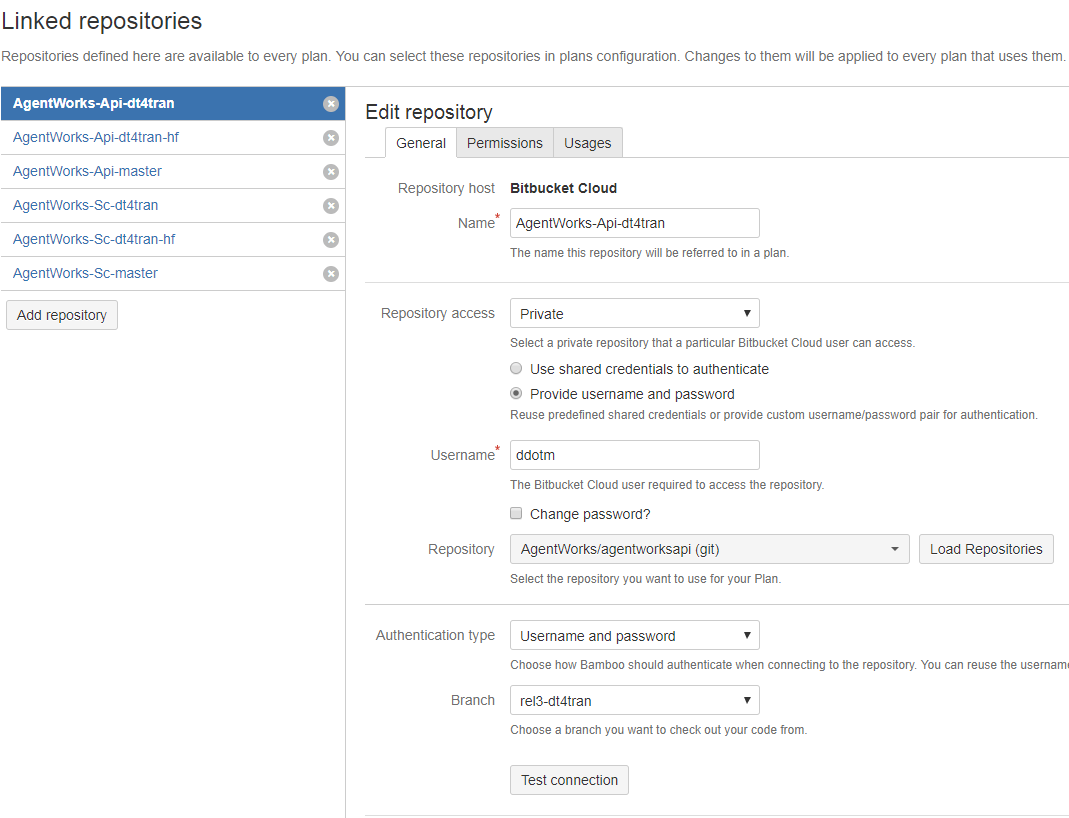
The name is prefixed with the system name — AgentWorks

The system name is followed by the application name — Api (for Web API) or Sc (for static content)

The application name is followed by the release name part of the code branch — dt4tran for DT4 transactional, dt4tran-hf for DT4 transactional hotfix

##### Linked Repositories general configurations

Fig. 6.1 – Linked Repositories General configurations



Name – this is the Linked Repository name, composed as defined [here](#_Linked_Repositories_naming).

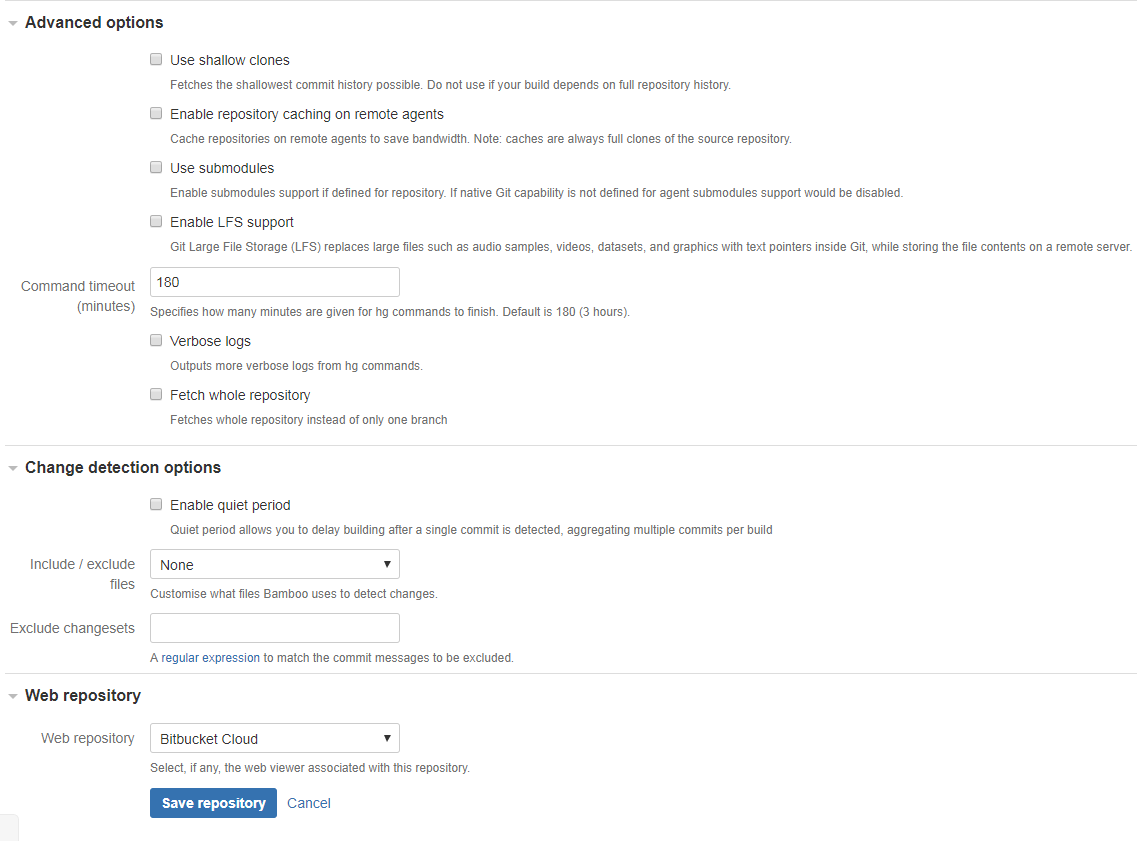
Repository Access — set to username and password. Should use AgentWorks service account.

Repository — set to the BitBucket repository name

Authentication type — Username and password

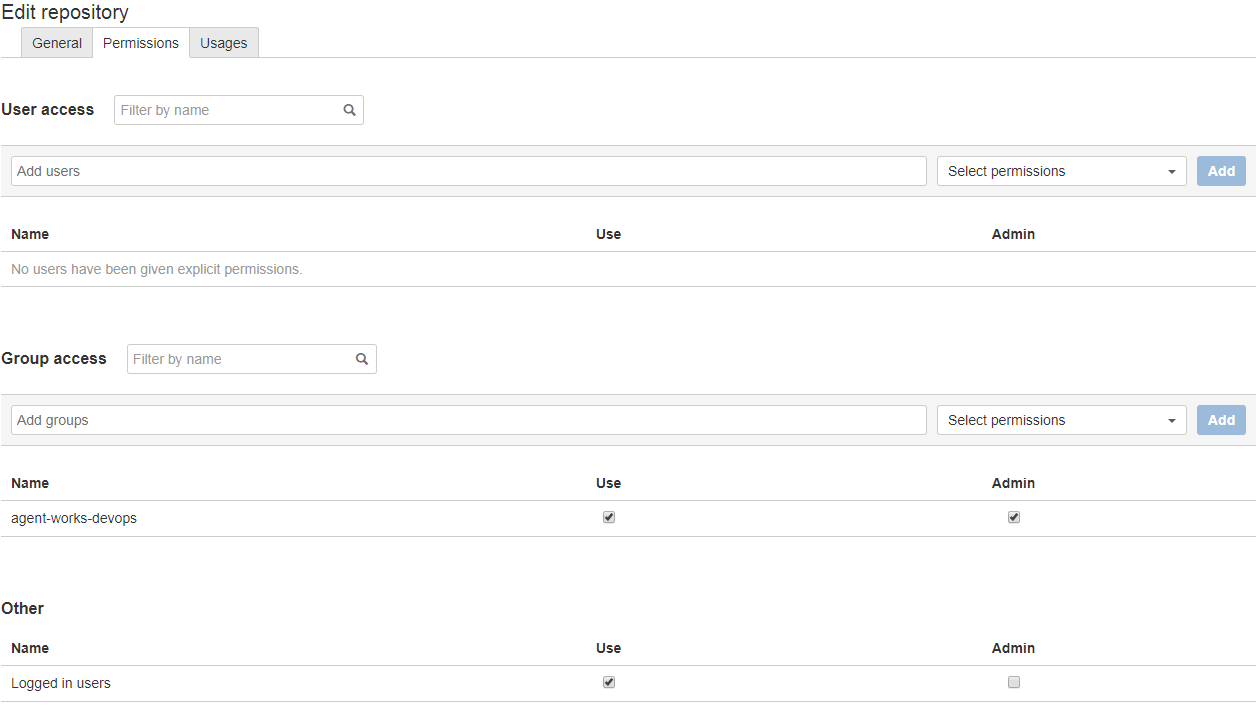
Branch — the branch name. Should match the branch name part of the linked repository name

Fig. 6.1.1 – Linked Repositories General configurations – Advanced, Change detection, and Web repository settings



The only important setting here is Web repository must be set to BitBucket Cloud.

Fig. 6.1.2 – Linked Repositories Permissions configurations



Add agent-works-devops group Use and Admin rights.

Remove all personal rights.

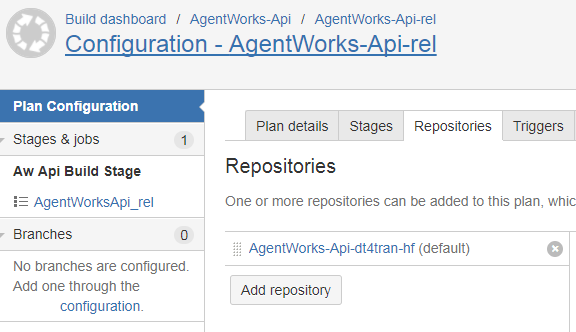
Fig. 6.1.3 – Linked Repositories Usages tab



This tab shows you which build plans use the selected linked repository. If no build plans use it, you should delete it by clicking the  symbol to the right of the linked repository name.

### Build Plan configuration — Repositories tab

Fig. 6.2 — Build plan repositories tab

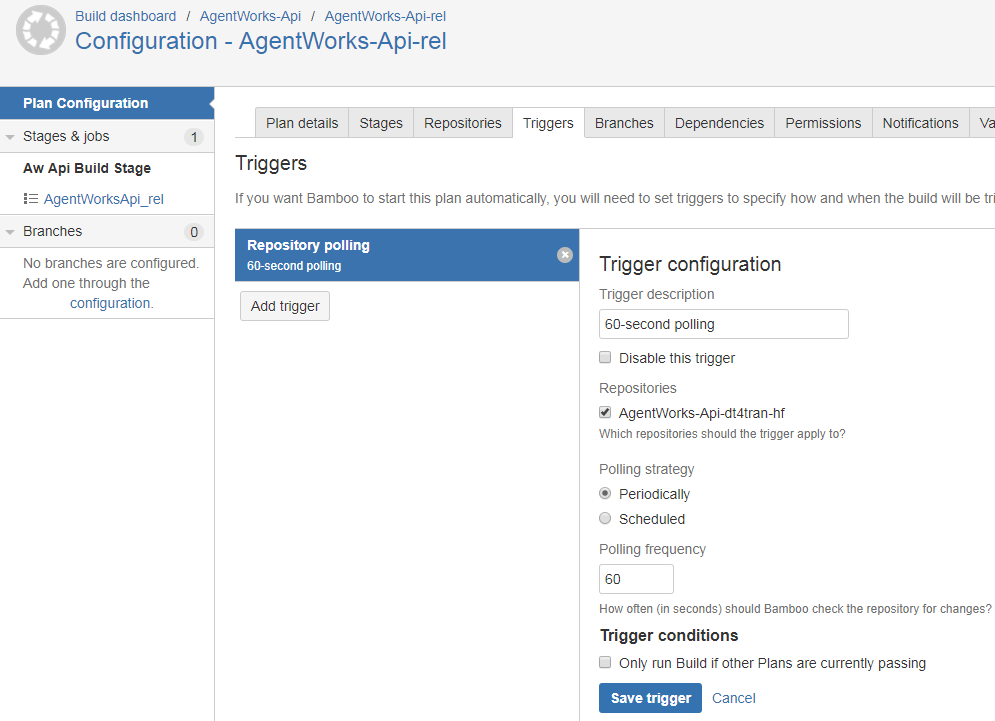


This is where a [linked repository](#_Linked_Repository_naming) is made available for use by tasks of the build plan.

So, at the time of the writing, AgentWorks-Api-rel build plan pointed to AgentWorks-Api-dt4tran-hf linked repository, which points to rel3-dt4tran-hf branch of agentworksapi BitBucket repo.

### Build Plan configuration — Triggers tab

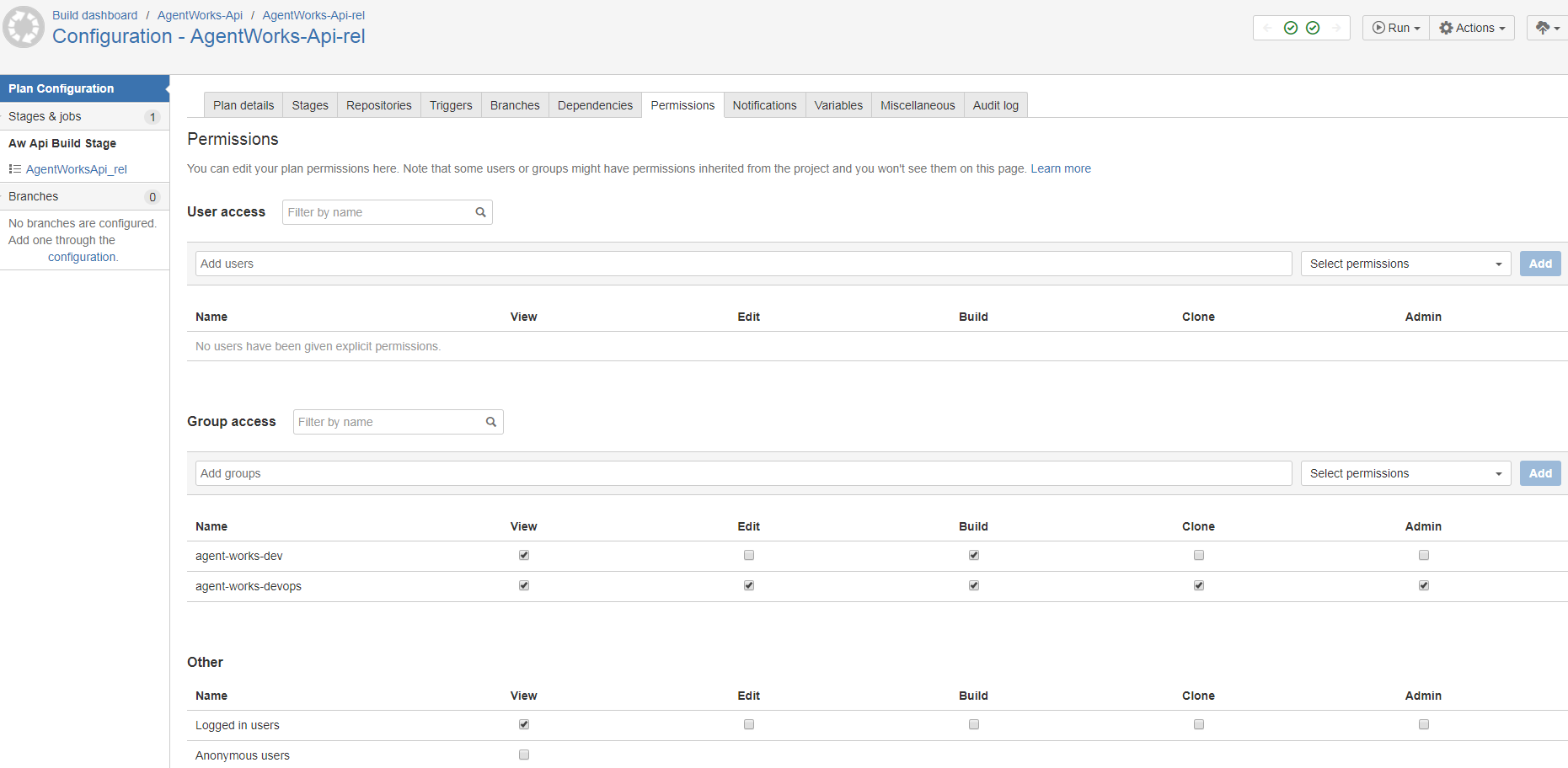
Fig. 6.3 — Build plan Triggers tab



AgentWorks build plans are set to poll the linked repository at 60-second intervals and be triggered by repository changes

### Build Plan configuration — Permissions tab

Fig. 6.4 — Build plan Permissions tab



Permissions tab governs View, Edit, Build, Clone, and Admin access to the build plan.

agent-works-dev group should include all developers on the team. This group can view and start a build.

agent-works-devops group has full permissions and should include only the people responsible for managing and administering DevOps for the team.

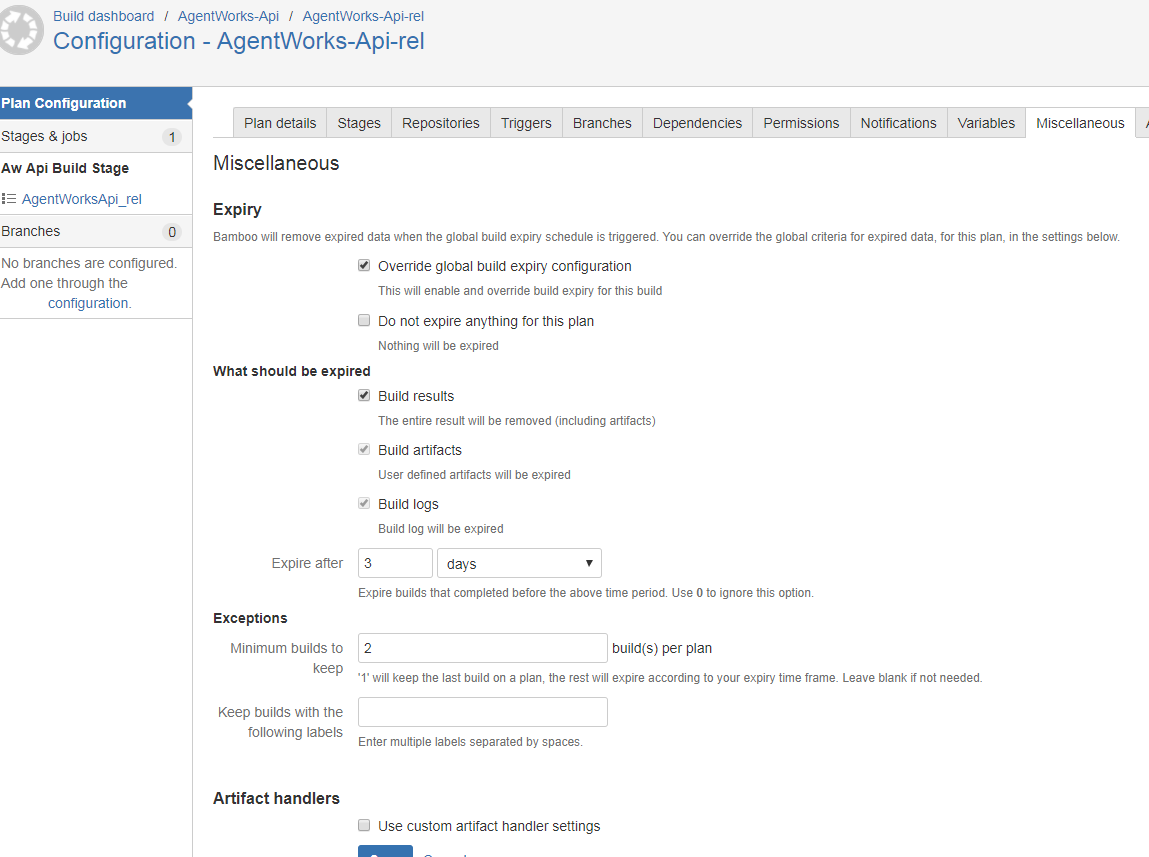
Logged in users can view the build.

Anonymous users have no permissions.

Avoid granting permissions at a User level. Rather add users to existing groups or create a new group if existing groups’ permissions do not fit a use case.

### Build Plan configuration — Miscellaneous tab

Fig. 6.5 — Build plan Miscellaneous tab



Miscellaneous tab is where expiry rules are configured.

AgentWorks builds artifacts are deleted after 3 days, with the exception of the last 2 builds. That should be sufficient to provide for the latest-deploy and rollback use cases. If requirements change, modify the settings accordingly.

## Bamboo deploys configuration, maintenance, administration, and management

Bamboo Deploy projects role is to deploy build artifacts to a specified environment. Each hosting environment has a corresponding deploy project.

Deploy project list can be accessed at <http://172.28.114.109:8085/deploy/viewAllDeploymentProjects.action>

Fig. 7.1 — deploy projects list view



AgentWorks has been granted permission to deploy to D2-3-5 and Q2-3-5 environments. The AgentWorks Dev Server also has a number of defined IIS hosted environments.

### Deploy Project and Environment naming conventions

Environment name consists of 3 parts separated by spaces.

Example - D2 Aw Api

D2 is the target hosting environment

Aw is the system name, AgentWorks

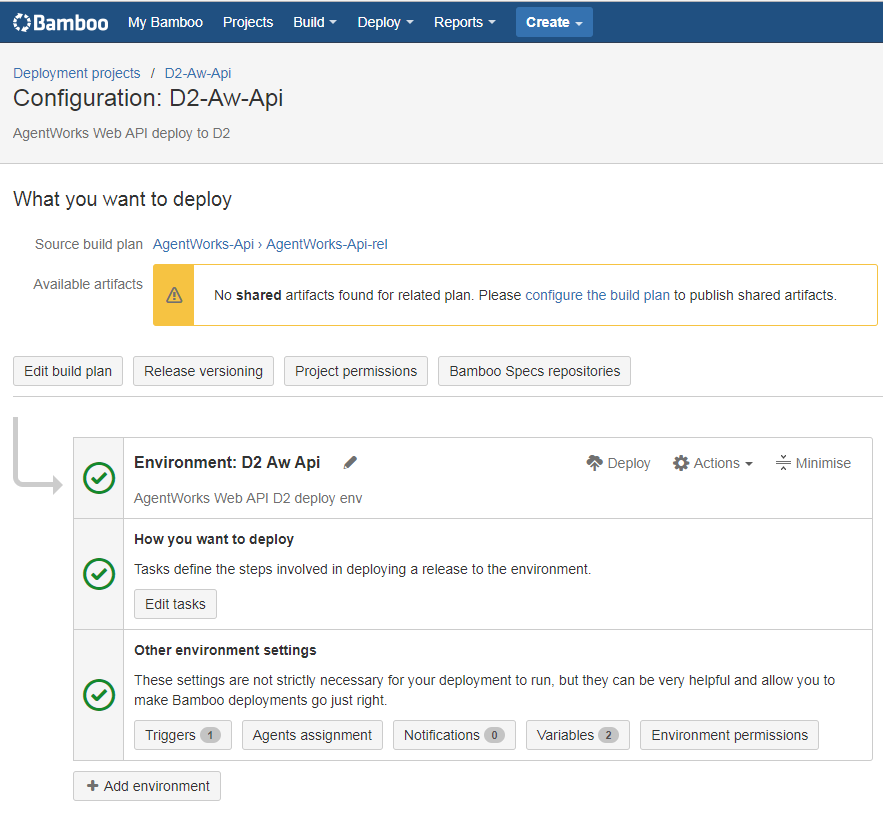
Api is the subsystem name, Web API

Deploy project name is composed the same, except the parts are linked by hyphens instead of spaces.

Example — D2-Aw-Api

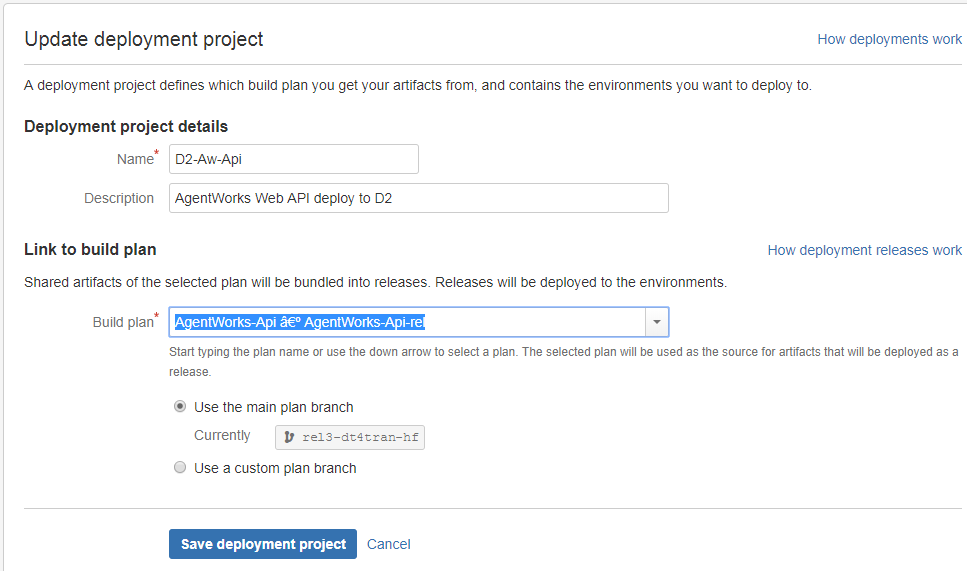
### Deploy Project configuration

Fig. 7.2 — Deploy project configuration



Edit build plan button will navigate to the screen where the deploy can be assigned to a specific build plan.

Fig. 7.3 — Deploy — Edit build plan



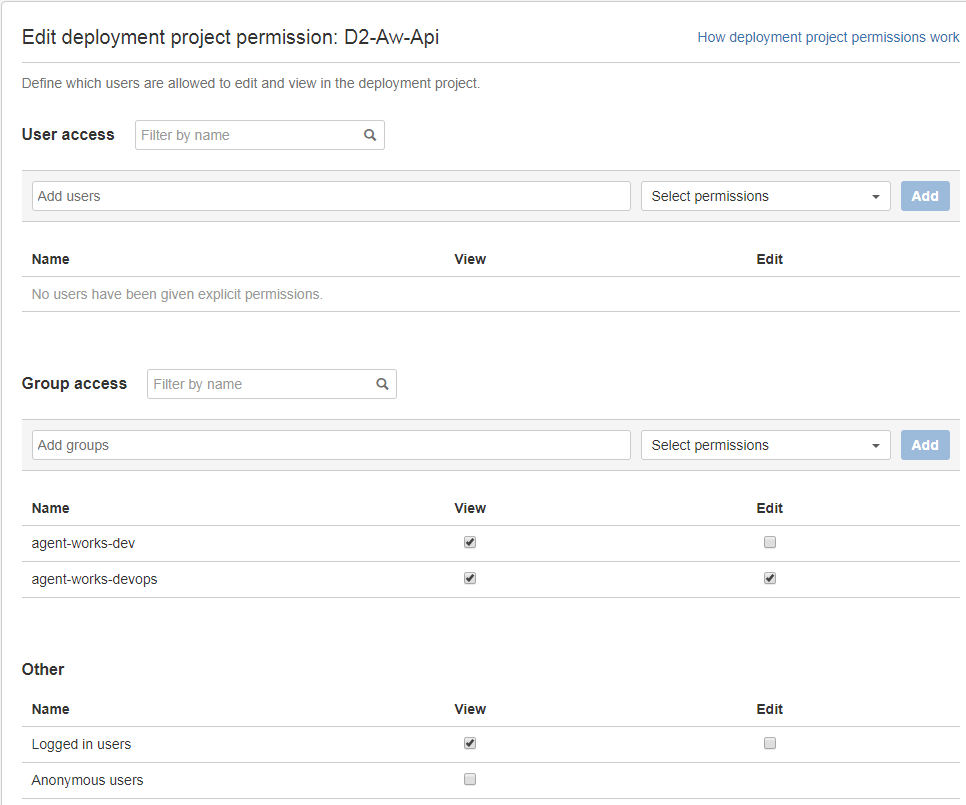
All defined build plans are available in the Build plan dropdown. Selecting a specific build plan will accomplish the deployment of artifacts of the selected build to the deploy’s target hosting environment.

In this example, AgentWorks Web API release branch build artifacts will be deployed to AgentWorks D2 Web API hosting environment.

The source code branch that the build is building from is displayed below the build plan, rel3-dt4tran-hf in this example.

### Deploy projects permissions

Fig. 7.4 — Deploy — project permissions



The same rules apply as for the [build plan permissions](#_Build_Plan_configuration).

DevOps group gets all rights; Dev gets View only.

### Deploy triggers settings

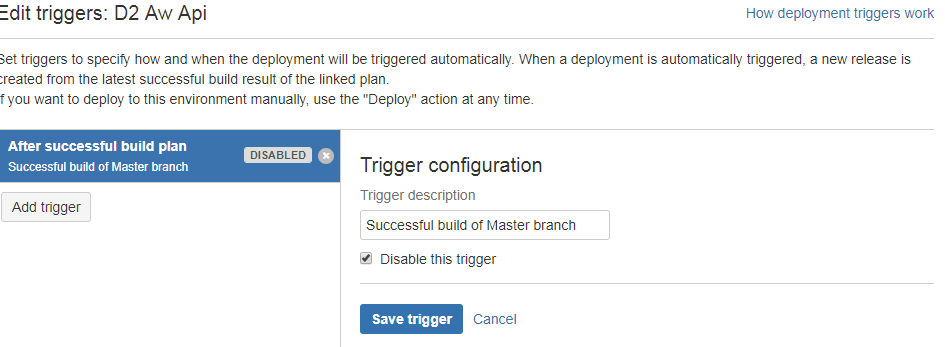
Fig. 7.5 — Deploy — triggers



AgentWorks deploys are triggered by successful completions of associated builds.

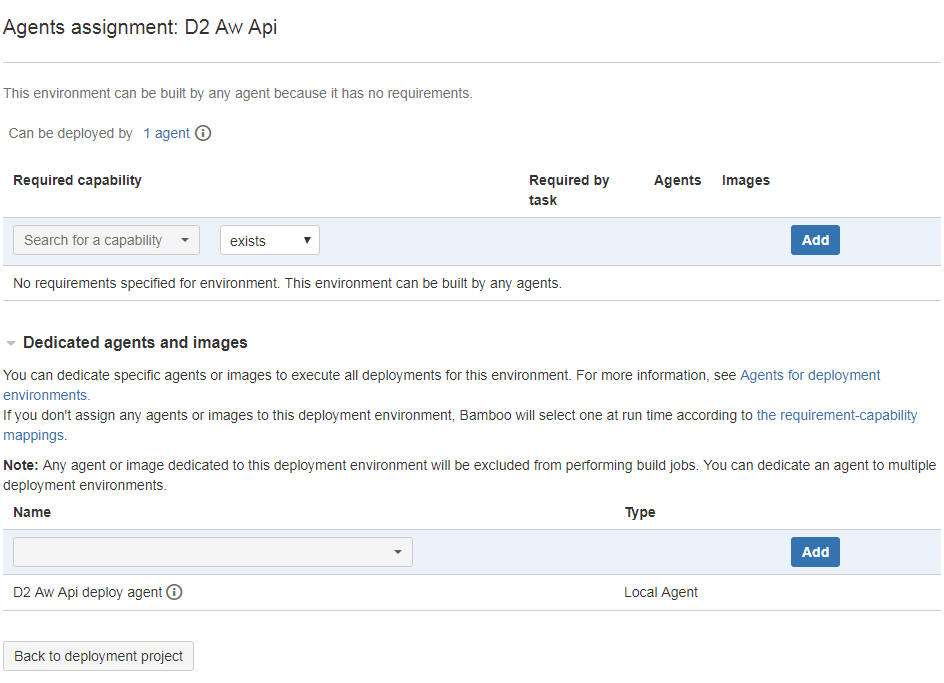
If a certain deploy needs to be disabled for a period of time, the easiest and least invasive way to accomplish that is to disable the deploy trigger.

Fig. 7.6 — Deploy — Disabled trigger



### Deploy Agents assignment

Fig. 7.7 — Deploy — Agents assignment



An agent is a service that executes Bamboo builds and deployments.

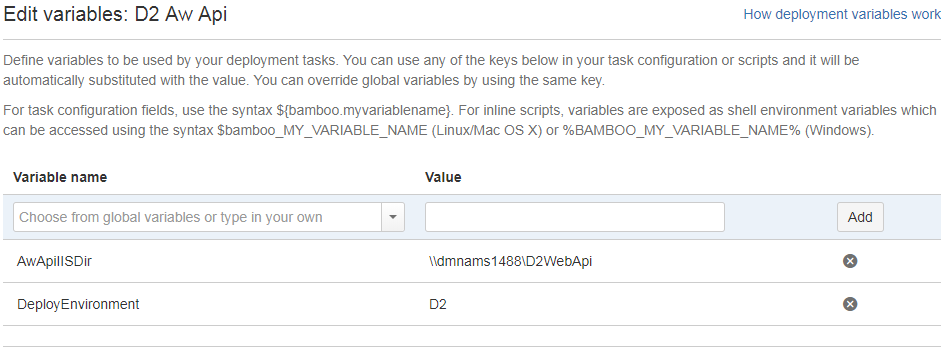
AgentWorks Bamboo plan includes unlimited local agents.

Each agent can run only one job at a time, so AgentWorks has a local agent dedicated for each build and deploy plan. This makes sure that builds and deploys can run concurrently.

Agents can be assigned to deploys on this tab or on the Agents summary page <http://172.28.114.109:8085/admin/agent/configureAgents!doDefault.action>

### Deploy variables

Fig. 7.8 — Deploy — Variables



Deploys define 2 variables each.

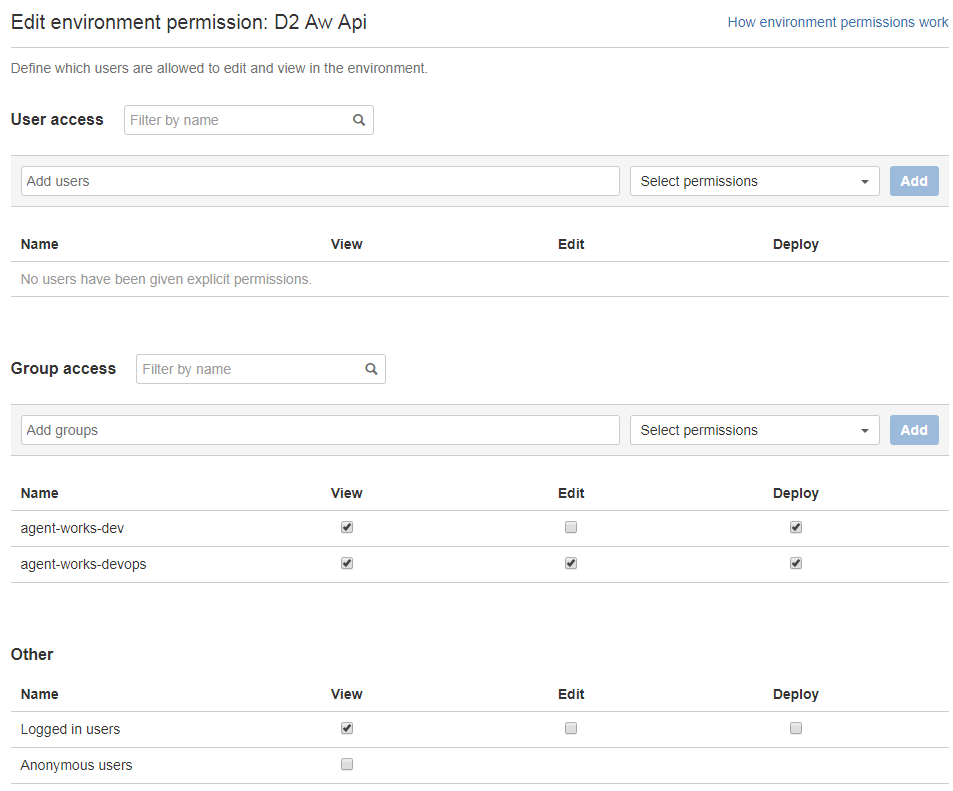
AwScIISDir or AwApiIISDir — set to the share directory path, e.g. \\dmnams1488\D2WebApi

DeployEnvironment — the name of the deploy environment, must match the names of config files or folders, e.g. D2

IMPORTANT: DeployEnvironment value must exactly match (case insensitive) the config directory for the target environment.

### Deploy Environment permissions

Fig. 7.9.1 — Deploy — Environment permissions for Dev server and D environments



agent-works-dev – view, deploy

agent-works-devops — view, edit, deploy

Logged in users — view

Fig. 7.9.2 — Deploy — Environment permissions for Q environments



agent-works-dev – view

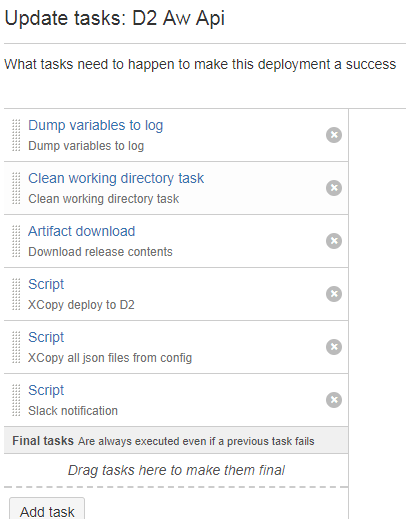
agent-works-devops — view, edit, deploy

agent-works-qs — view, deploy

Logged in users — view

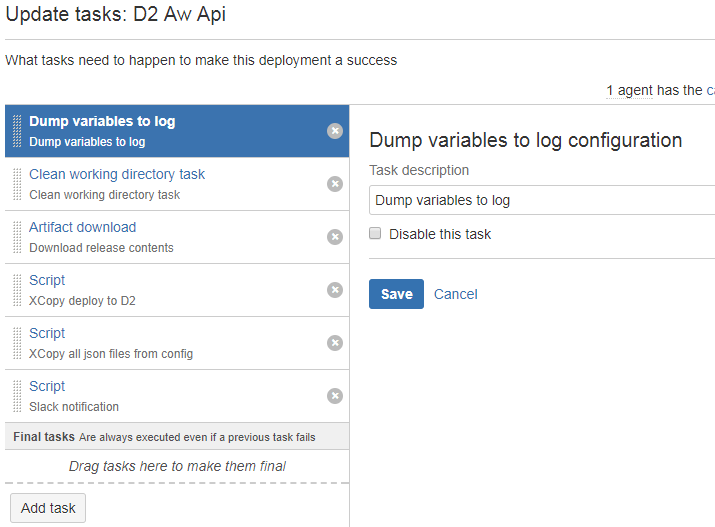
### API Deploy tasks

Fig. 7.10 — API Deploy Tasks list



#### API Deploy task — Dump variables to log configuration

Fig. 7.10.1 — Dump variables to log task configuration



##### Sample variable log dump

simple 17-Nov-2017 09:58:21 ---- DUMPING VARIABLES TO LOG ---

simple 17-Nov-2017 09:58:21 key: [AwApiBuildDrop] value: [D:\AwBuildDrop\AwApi] type: GLOBAL

simple 17-Nov-2017 09:58:21 key: [AwApiD5FtpConnectString] value: [ftp://[UID]:[PWD]@[TARGET IP]] type: GLOBAL

simple 17-Nov-2017 09:58:21 key: [AwApiD5RemoteDir] value: [D:\DeployTarget\D5\AWAPI] type: GLOBAL

simple 17-Nov-2017 09:58:21 key: [AwApiIISDir] value: [\\dmnams1488\D2WebApi] type: ENVIRONMENT

simple 17-Nov-2017 09:58:21 key: [AwApiSolutionFileName] value: [AwApi.sln] type: GLOBAL

simple 17-Nov-2017 09:58:21 key: [AwScBuildDrop] value: [D:\AwBuildDrop\AwSc] type: GLOBAL

simple 17-Nov-2017 09:58:21 key: [AwScD5RemoteDir] value: [D:\DeployTarget\D5\AWSC] type: GLOBAL

simple 17-Nov-2017 09:58:21 key: [DependencyTriggerReason.triggeringBuildResultKey] value: [AWAPI-AWAPIR-20] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [DeployEnvironment] value: [D2] type: ENVIRONMENT

simple 17-Nov-2017 09:58:21 key: [GlobalNodeModulesDir] value: [C:\Users\fcbamboobuild\AppData\Roaming\npm\node\_modules] type: GLOBAL

simple 17-Nov-2017 09:58:21 key: [NugetExeFullPath] value: ["C:\Program Files (x86)\NuGet\nuget.exe"] type: GLOBAL

simple 17-Nov-2017 09:58:21 key: [agentId] value: [33193985] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [agentWorkingDirectory] value: [D:\bamboo-home\xml-data\build-dir] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [build.working.directory] value: [D:\bamboo-home\xml-data\build-dir\11239425-11304961] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [buildNumber] value: [20] type: VERSION

simple 17-Nov-2017 09:58:21 key: [buildResultKey] value: [AWAPI-AWAPIR-20] type: VERSION

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.command.MSTest] value: [C:\Program Files (x86)\MSTest\MSTest.exe] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.command.WinSCP.com] value: [C:\Program Files (x86)\winscp577\WinSCP.com] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.command.curl] value: [C:\Program Files\curl] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.command.nuget] value: [C:\Program Files (x86)\NuGet\NuGet.exe] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.devenv.MSTest] value: [C:\Program Files (x86)\MSTest\MSTest.exe] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.msbuild.MSBuild 14] value: [C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.msbuild.MSBuild 15] value: [C:\Program Files (x86)\Microsoft Visual Studio\2017\BuildTools\MSBuild\15.0\Bin\MSBuild.exe] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.msbuild.MSBuild v14.0 (32bit)] value: [C:\Program Files (x86)\MSBuild\14.0\bin\MSBuild.exe] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.msbuild.MSBuild v14.0 (64bit)] value: [C:\Program Files (x86)\MSBuild\14.0\bin\amd64\MSBuild.exe] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.msbuild.MSBuild v4.0 (32bit)] value: [C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.msbuild.MSBuild v4.0 (64bit)] value: [C:\Windows\Microsoft.NET\Framework64\v4.0.30319\MSBuild.exe] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.builder.node.Node.js] value: [C:\Program Files\nodejs\node.exe] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.git.executable] value: [C:\Program Files\Git\cmd\git.exe] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.jdk.JDK] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.jdk.JDK 1.8] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [capability.system.jdk.JDK 1.8.0\_92] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [deploy.environment] value: [D2 Aw Api] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [deploy.project] value: [AgentWorks-Api-D2] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [deploy.release] value: [release-209] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [deploy.release.previous] value: [release-208] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [deploy.rollback] value: [false] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [deploy.version] value: [release-209] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [deploy.version.previous] value: [release-208] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [planKey] value: [AWAPI-AWAPIR] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planName] value: [AgentWorks-Api - AgentWorks-Api-rel] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.1.branch] value: [rel3-dt4tran] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.1.branchDisplayName] value: [rel3-dt4tran] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.1.branchName] value: [rel3-dt4tran] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.1.name] value: [AgentWorks-Api-dt4tran] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.1.repositoryUrl] value: [https://bitbucket.org/AgentWorks/agentworksapi] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.1.revision] value: [d0ad9dc53e4b7e09457207927472519bfc117f05] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.1.type] value: [bbCloud] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.1.username] value: [ddotm] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.branch] value: [rel3-dt4tran] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.branchDisplayName] value: [rel3-dt4tran] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.branchName] value: [rel3-dt4tran] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.name] value: [AgentWorks-Api-dt4tran] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.repositoryUrl] value: [https://bitbucket.org/AgentWorks/agentworksapi] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.revision] value: [d0ad9dc53e4b7e09457207927472519bfc117f05] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.type] value: [bbCloud] type: VERSION

simple 17-Nov-2017 09:58:21 key: [planRepository.username] value: [ddotm] type: VERSION

simple 17-Nov-2017 09:58:21 key: [resultsUrl] value: [http://172.28.114.109:8085/deploy/viewDeploymentResult.action?deploymentResultId=35094750] type: CUSTOM

simple 17-Nov-2017 09:58:21 key: [shortPlanKey] value: [AWAPIR] type: VERSION

simple 17-Nov-2017 09:58:21 key: [shortPlanName] value: [AgentWorks-Api-rel] type: VERSION

simple 17-Nov-2017 09:58:21 key: [working.directory] value: [D:\bamboo-home\xml-data\build-dir\11239425-11304961] type: CUSTOM

simple 17-Nov-2017 09:58:21 ---- END DUMPING VARIABLES TO LOG ---

#### API Deploy task — Clean working directory task configuration

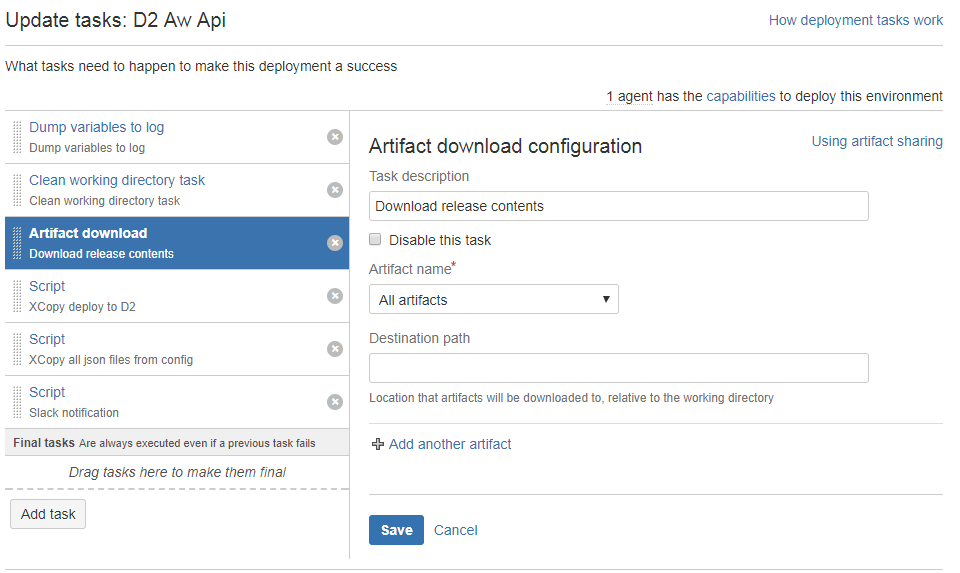
Fig. 7.10.2 — Clean working directory task configuration



This task deletes all files and directories from the working directory

#### API Deploy task — Artifact download task configuration

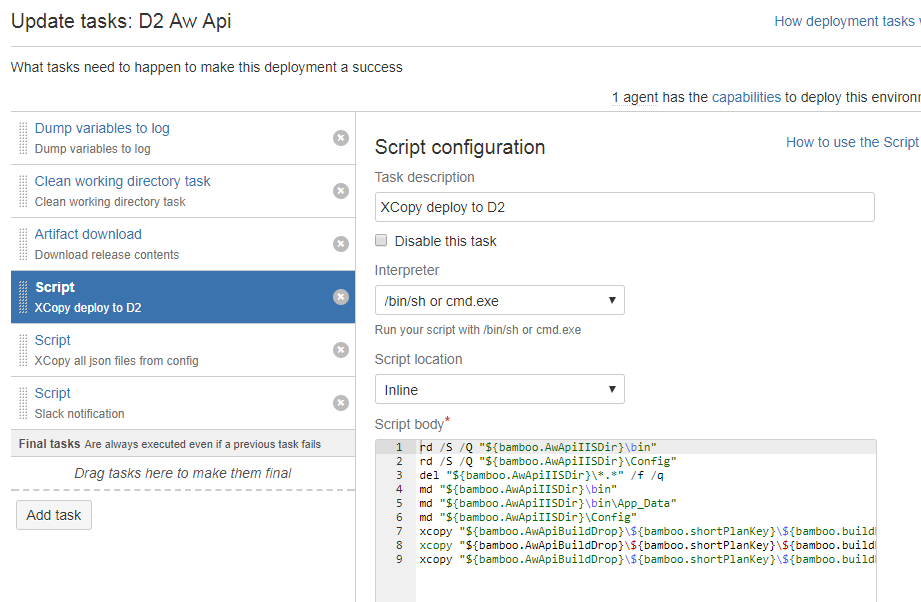
Fig. 7.10.2 — Artifact download task configuration



This task moved and stages build artifacts for deploy.

#### API Deploy task — Script — XCopy deploy task configuration

Fig. 7.10.3 — Script — XCopy Deploy task configuration



Script body

rd /S /Q "${bamboo.AwApiIISDir}\bin"

rd /S /Q "${bamboo.AwApiIISDir}\Config"

del "${bamboo.AwApiIISDir}\\*.\*" /f /q

md "${bamboo.AwApiIISDir}\bin"

md "${bamboo.AwApiIISDir}\bin\App\_Data"

md "${bamboo.AwApiIISDir}\Config"

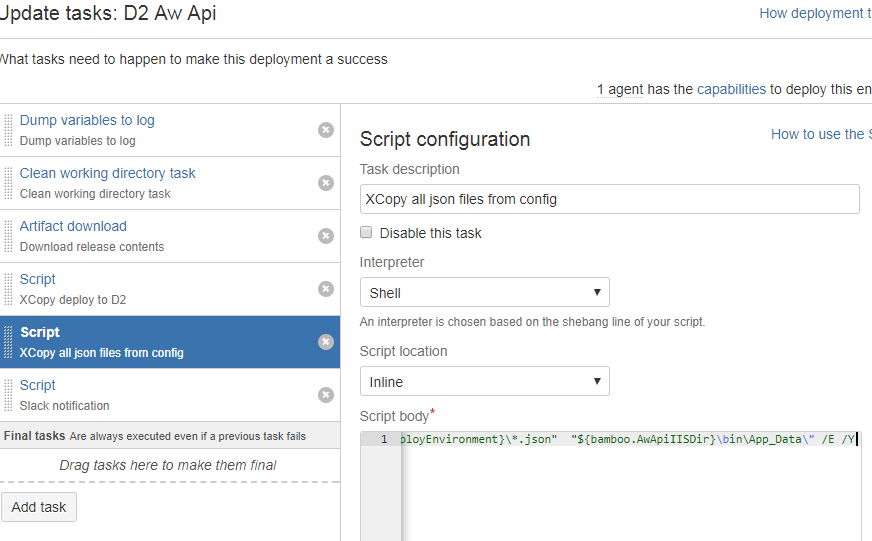
xcopy "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\bin\\*.\*" "${bamboo.AwApiIISDir}\bin\" /E

xcopy "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\WebConfigs\${bamboo.DeployEnvironment}\Web.config" "${bamboo.AwApiIISDir}\" /E

xcopy "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\Config\${bamboo.DeployEnvironment}\\*.\*" "${bamboo.AwApiIISDir}\Config\" /E

#### API Deploy task — Script — XCopy deploy task configuration

Fig. 7.10.4 — Script — XCopy all json files from config task configuration



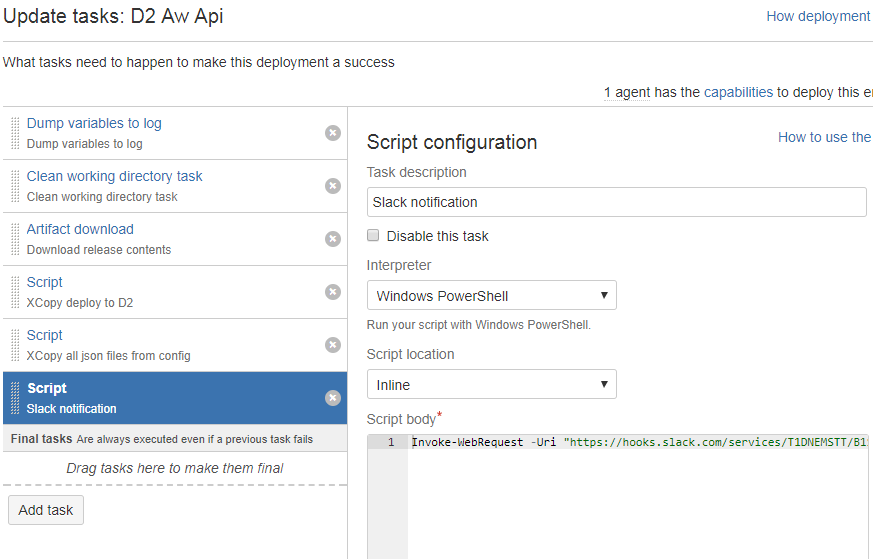
This task deploys config files from the environment specific config directory.

Script body

xcopy "${bamboo.AwApiBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\Config\${bamboo.DeployEnvironment}\\*.json" "${bamboo.AwApiIISDir}\bin\App\_Data\" /E /Y

#### API Deploy task — Script — Slack notification task configuration

Fig. 7.10.5 — Script — Slack notification task configuration



Interpreter: Windows PowerShell

Script body

$AllProtocols = [System.Net.SecurityProtocolType]'Ssl3,Tls,Tls11,Tls12'

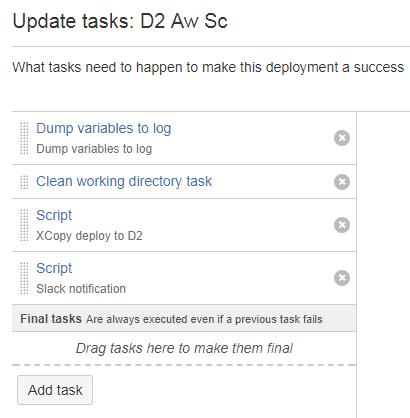
[System.Net.ServicePointManager]::SecurityProtocol = $AllProtocols

Invoke-WebRequest -Uri "https://hooks.slack.com/services/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" -Body "{""color"": ""#2ab27b"", ""text"": ""$${bamboo.planRepository.name} \*DEPLOY\* notification: ${bamboo.planRepository.branchName} build ${bamboo.buildResultKey} \*SUCCESSFULLY\* deployed to \*${bamboo.DeployEnvironment}\*. View results at <${bamboo.resultsUrl}|${bamboo.resultsUrl}>""}" -ContentType "application/json" -Method Post -UseBasicParsing

### Static Content Deploy tasks

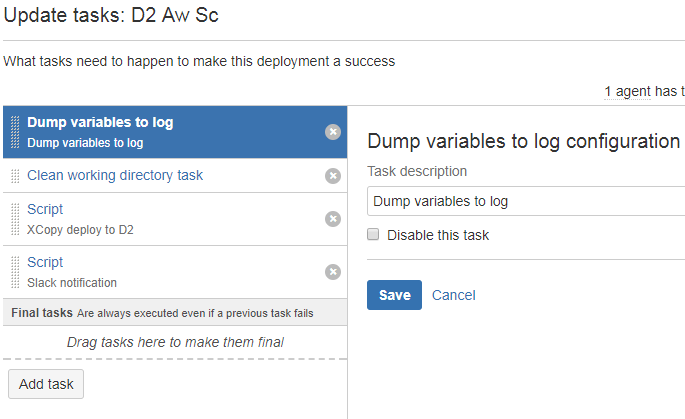
#### Static Content Deploy task list

Fig. 7.11 — Static Content deploy task list



#### Static Content Deploy task — Dump variables to log task configuration

Fig. 7.11.1 — Dump variables to log task configuration



##### Sample log dump

simple 10-Nov-2017 12:30:08 ---- DUMPING VARIABLES TO LOG ---

simple 10-Nov-2017 12:30:08 key: [AwApiBuildDrop] value: [D:\AwBuildDrop\AwApi] type: GLOBAL

simple 10-Nov-2017 12:30:08 key: [AwApiD5FtpConnectString] value: [ftp://[UID]:[PWD]@[TARGET IP]] type: GLOBAL

simple 10-Nov-2017 12:30:08 key: [AwApiD5RemoteDir] value: [D:\DeployTarget\D5\AWAPI] type: GLOBAL

simple 10-Nov-2017 12:30:08 key: [AwApiSolutionFileName] value: [AwApi.sln] type: GLOBAL

simple 10-Nov-2017 12:30:08 key: [AwScBuildDrop] value: [D:\AwBuildDrop\AwSc] type: GLOBAL

simple 10-Nov-2017 12:30:08 key: [AwScD5RemoteDir] value: [D:\DeployTarget\D5\AWSC] type: GLOBAL

simple 10-Nov-2017 12:30:08 key: [AwScIISDir] value: [\\dmnams1488\D2StaticContent] type: ENVIRONMENT

simple 10-Nov-2017 12:30:08 key: [DependencyTriggerReason.triggeringBuildResultKey] value: [AWSC-AWSCR-17] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [DeployEnvironment] value: [D2] type: ENVIRONMENT

simple 10-Nov-2017 12:30:08 key: [GlobalNodeModulesDir] value: [C:\Users\fcbamboobuild\AppData\Roaming\npm\node\_modules] type: GLOBAL

simple 10-Nov-2017 12:30:08 key: [NugetExeFullPath] value: ["C:\Program Files (x86)\NuGet\nuget.exe"] type: GLOBAL

simple 10-Nov-2017 12:30:08 key: [agentId] value: [163841] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [agentWorkingDirectory] value: [D:\bamboo-home\xml-data\build-dir] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [build.working.directory] value: [D:\bamboo-home\xml-data\build-dir\11239426-11304962] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [buildNumber] value: [17] type: VERSION

simple 10-Nov-2017 12:30:08 key: [buildResultKey] value: [AWSC-AWSCR-17] type: VERSION

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.command.MSTest] value: [C:\Program Files (x86)\MSTest\MSTest.exe] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.command.WinSCP.com] value: [C:\Program Files (x86)\winscp577\WinSCP.com] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.command.curl] value: [C:\Program Files\curl] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.command.nuget] value: [C:\Program Files (x86)\NuGet\NuGet.exe] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.devenv.MSTest] value: [C:\Program Files (x86)\MSTest\MSTest.exe] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.msbuild.MSBuild 14] value: [C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.msbuild.MSBuild 15] value: [C:\Program Files (x86)\Microsoft Visual Studio\2017\BuildTools\MSBuild\15.0\Bin\MSBuild.exe] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.msbuild.MSBuild v14.0 (32bit)] value: [C:\Program Files (x86)\MSBuild\14.0\bin\MSBuild.exe] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.msbuild.MSBuild v14.0 (64bit)] value: [C:\Program Files (x86)\MSBuild\14.0\bin\amd64\MSBuild.exe] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.msbuild.MSBuild v4.0 (32bit)] value: [C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.msbuild.MSBuild v4.0 (64bit)] value: [C:\Windows\Microsoft.NET\Framework64\v4.0.30319\MSBuild.exe] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.builder.node.Node.js] value: [C:\Program Files\nodejs\node.exe] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.git.executable] value: [C:\Program Files\Git\cmd\git.exe] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.jdk.JDK] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.jdk.JDK 1.8] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [capability.system.jdk.JDK 1.8.0\_92] value: [C:\Program Files\Java\jre1.8.0\_92] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [deploy.environment] value: [D2 Aw Static Content] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [deploy.project] value: [AgentWorks-SC-D2] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [deploy.release] value: [release-322] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [deploy.release.previous] value: [release-321] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [deploy.rollback] value: [false] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [deploy.version] value: [release-322] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [deploy.version.previous] value: [release-321] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [planKey] value: [AWSC-AWSCR] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planName] value: [AgentWorks-StaticContent - AgentWorks-Sc-rel] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.1.branch] value: [rel3-dt4tran] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.1.branchDisplayName] value: [rel3-dt4tran] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.1.branchName] value: [rel3-dt4tran] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.1.name] value: [AgentWorks-Sc-dt4tran] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.1.repositoryUrl] value: [https://bitbucket.org/AgentWorks/agentworksstaticcontent] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.1.revision] value: [75007c43b78f967aca7448f7cf46fbd16fedd217] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.1.type] value: [bbCloud] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.1.username] value: [ddotm] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.branch] value: [rel3-dt4tran] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.branchDisplayName] value: [rel3-dt4tran] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.branchName] value: [rel3-dt4tran] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.name] value: [AgentWorks-Sc-dt4tran] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.repositoryUrl] value: [https://bitbucket.org/AgentWorks/agentworksstaticcontent] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.revision] value: [75007c43b78f967aca7448f7cf46fbd16fedd217] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.type] value: [bbCloud] type: VERSION

simple 10-Nov-2017 12:30:08 key: [planRepository.username] value: [ddotm] type: VERSION

simple 10-Nov-2017 12:30:08 key: [resultsUrl] value: [http://172.28.114.109:8085/deploy/viewDeploymentResult.action?deploymentResultId=35094674] type: CUSTOM

simple 10-Nov-2017 12:30:08 key: [shortPlanKey] value: [AWSCR] type: VERSION

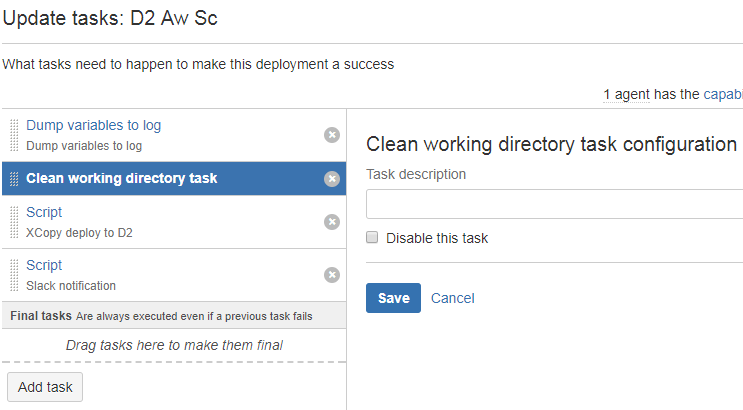
simple 10-Nov-2017 12:30:08 key: [shortPlanName] value: [AgentWorks-Sc-rel] type: VERSION

simple 10-Nov-2017 12:30:08 key: [working.directory] value: [D:\bamboo-home\xml-data\build-dir\11239426-11304962] type: CUSTOM

simple 10-Nov-2017 12:30:08 ---- END DUMPING VARIABLES TO LOG ---

#### Static Content Deploy task — Clean working directory task configuration

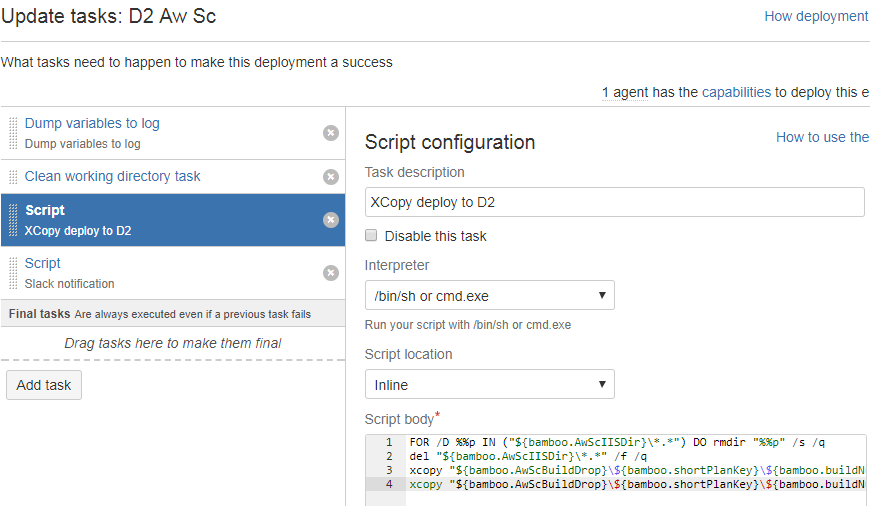
Fig. 7.11.1 — Clean working directory task configuration



This task will delete all files and directories under the working directory

#### Static Content Deploy task — Script — XCopy deploy task configuration

Fig. 7.11.1 — Script —XCopy deploy task configuration



Script body

FOR /D %%p IN ("${bamboo.AwScIISDir}\\*.\*") DO rmdir "%%p" /s /q

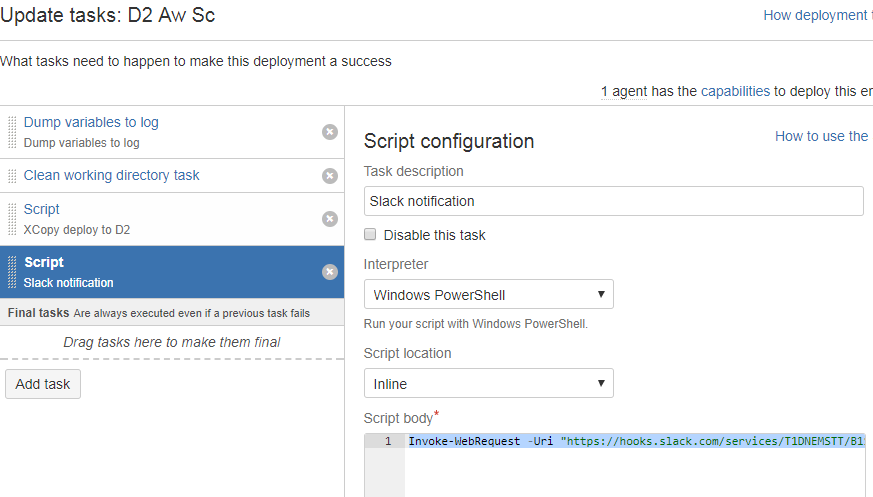
del "${bamboo.AwScIISDir}\\*.\*" /f /q

xcopy "${bamboo.AwScBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\app\\*.\*" "${bamboo.AwScIISDir}\" /E

xcopy "${bamboo.AwScBuildDrop}\${bamboo.shortPlanKey}\${bamboo.buildNumber}\configs\${bamboo.DeployEnvironment}\\*.\*" "${bamboo.AwScIISDir}\" /E

#### Static Content Deploy task — Script — Slack notification task configuration

Fig. 7.11.1 — Script —Slack notification task configuration



Interpreter: Windows PowerShell

Script body

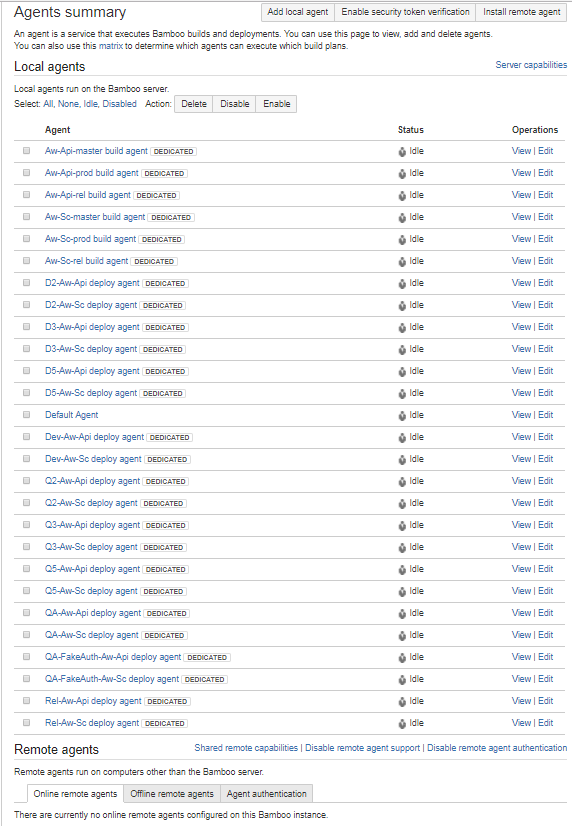
$AllProtocols = [System.Net.SecurityProtocolType]'Ssl3,Tls,Tls11,Tls12'

[System.Net.ServicePointManager]::SecurityProtocol = $AllProtocols

Invoke-WebRequest -Uri "https://hooks.slack.com/services/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" -Body "{""text"": ""${bamboo.planRepository.name} \*DEPLOY\* notification: ${bamboo.planRepository.branchName} build ${bamboo.buildResultKey} \*SUCCESSFULLY\* deployed to \*${bamboo.DeployEnvironment}\*. View results at <${bamboo.resultsUrl}|${bamboo.resultsUrl}>""}" -ContentType "application/json" -Method Post -UseBasicParsing

## Bamboo agents

Fig. 8.0 — AgentWorks build and deploy local agents list <http://172.28.114.109:8085/admin/agent/configureAgents!doDefault.action>



Note that each local agent except Default Agent is marked as dedicated.

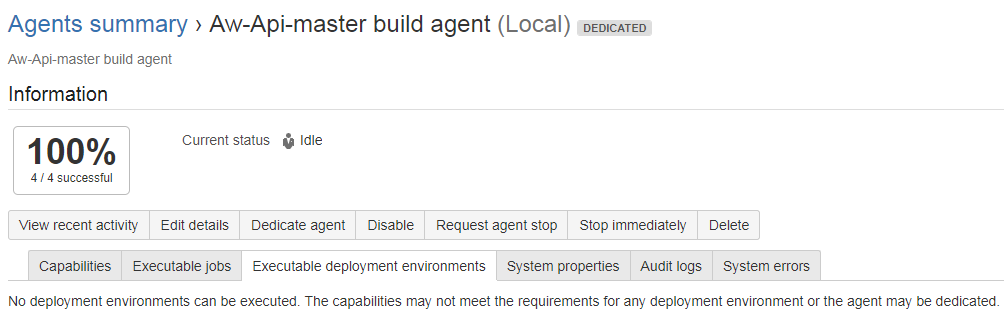
There is a dedicated agent defined for each build and deploy. That assures that the builds and deploys can run concurrently.

IMPORTANT: Part of defining a new build or deploy should be defining a new agent and dedicating it to run that build or deploy.

### Build agent configuration

Clicking on the View link for an agent in the list takes you to the agent summary page

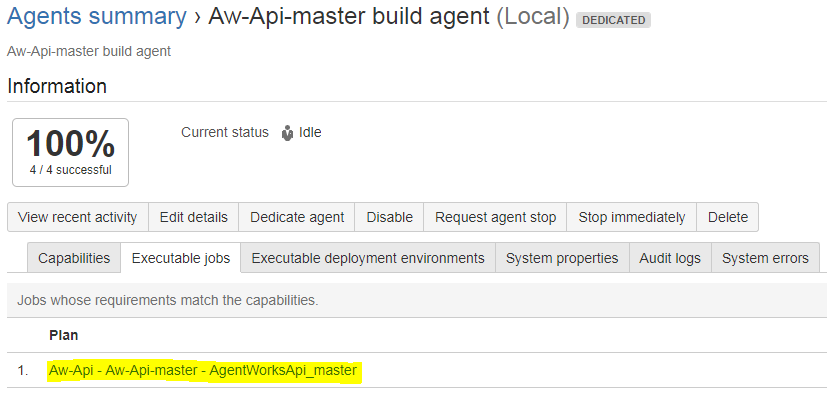
Fig. 9.0 — Build Agent summary



Build plan assignments are listed under the “Executable jobs” tab.

Deploy plan assignments are listed under the “Executable deployment environments” tab.

Fig. 9.1 — Build Agent — Executable jobs tab



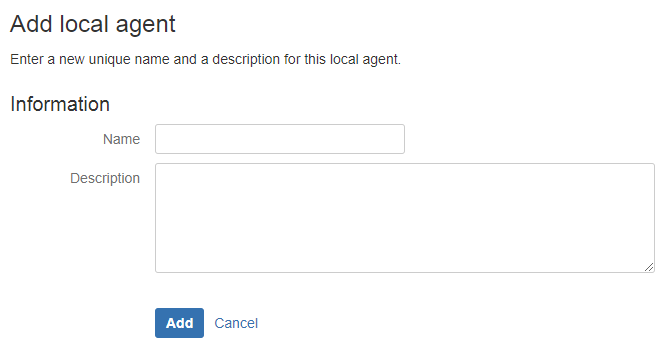
### Adding a new local agent

Each build and deploy plan should have a dedicated agent so jobs can execute concurrently.

To add a local agent, navigate to the configure agents page <http://172.28.114.109:8085/admin/agent/configureAgents!doDefault.action> and click on the “Add local agent” button.

That will take you to the Add local agent screen — <http://172.28.114.109:8085/admin/agent/addLocalAgent.action>

Fig. 10.0 — Add local agent screen



Fill in the name and optionally description.

The name should be composed as follows:

* Build or deploy plan name (e.g. Aw-Api-master)
* Space
* Build agent or deploy agent

Example of a build agent name: Aw-Api-master build agent

Example of a deploy agent name: D2-Aw-Api deploy agent

Once the name and description is in, click “Add” to add the agent.

### Assigning a new local agent to build or deploy

In the Agent view screen, click on the “Dedicate agent” tab

Fig. 11.1 — Build plans for a build agent

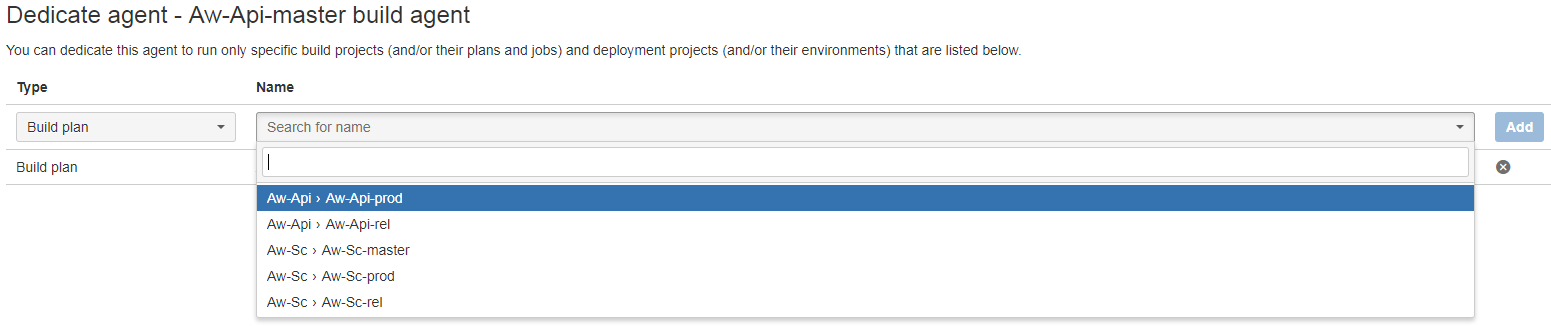
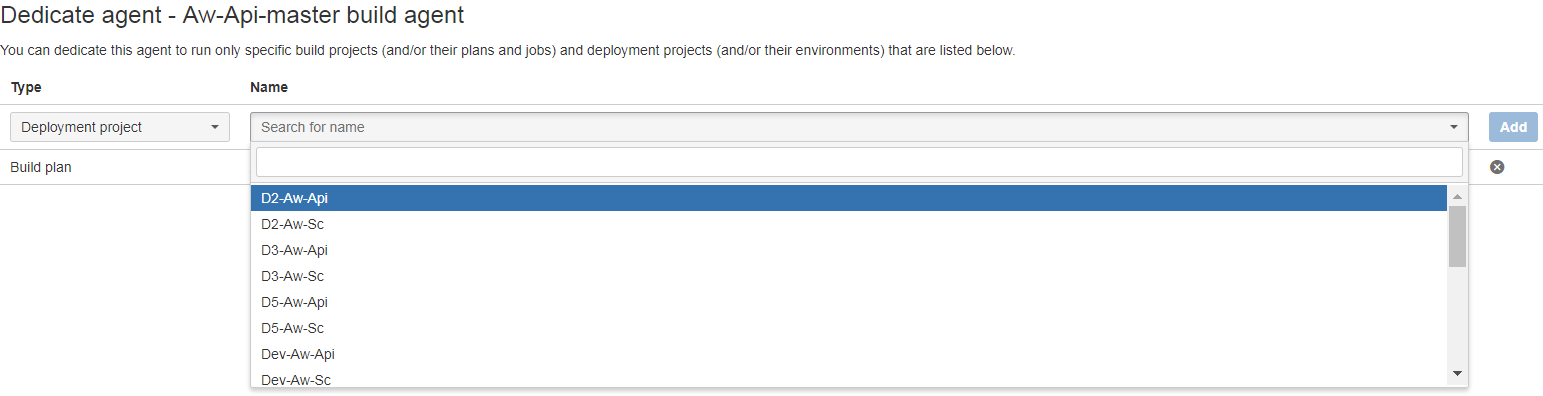


Fig. 11.2 — Deploy projects for a deploy agent



For Build plan agents, select “Build plan” under Type.

For Deploy plan agents, select

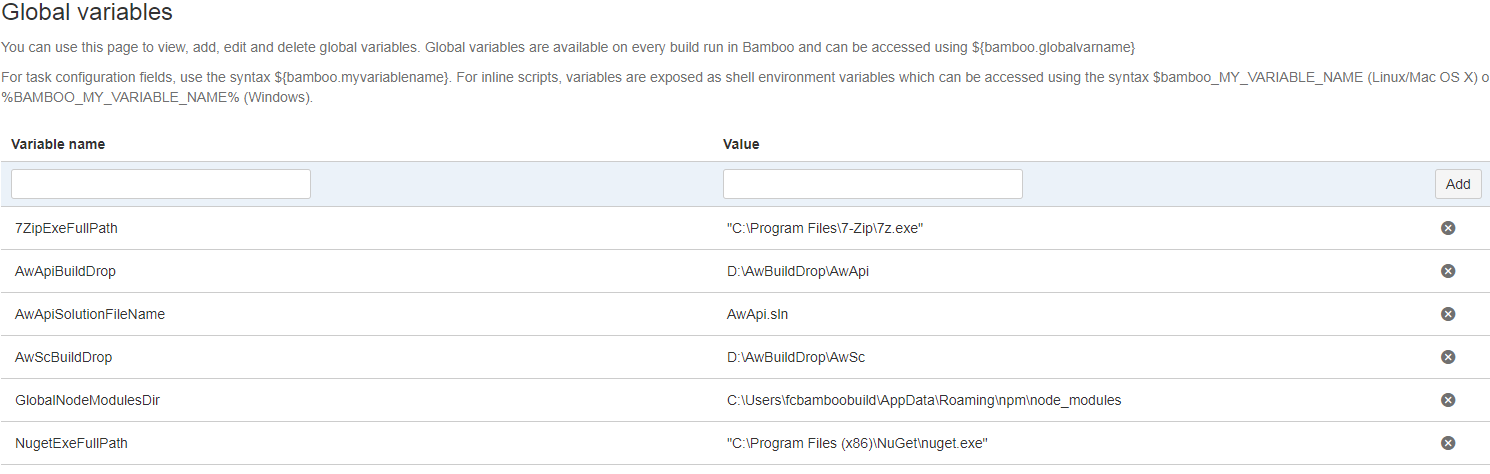
That will populate the Name dropdown with all defined plans of selected type.

Select the name of the build plan the agent was created for.

Click “Add”.

## Bamboo Global variables

Fig. 12 — Bamboo Global variables screen <http://172.28.114.109:8085/admin/configureGlobalVariables!doDefault.action>



Global variables should be used sparingly. Only variables that are truly reused by many or all plans should be defined globally.

Bamboo allows defining variables at plan/project level, and variables used for specific builds or deploys should be defined at plan/project level.

Globals are defined for executables paths to be used by command scripts and for build drop directories.

To use a global variable, like any other variable in Bamboo, use this syntax — ${bamboo.VARIABLE\_NAME}.

## Bamboo administration

### Groups

Fig. 13 — Bamboo Groups screen — <http://172.28.114.109:8085/admin/group/viewGroups.action>



Dev group should have the view and execute rights.

DevOps group should have the view, edit, and execute rights.

Qs group was created to give QA team members deploy rights to Q environments.

Bamboo-admin has all the right, so should be very limited.

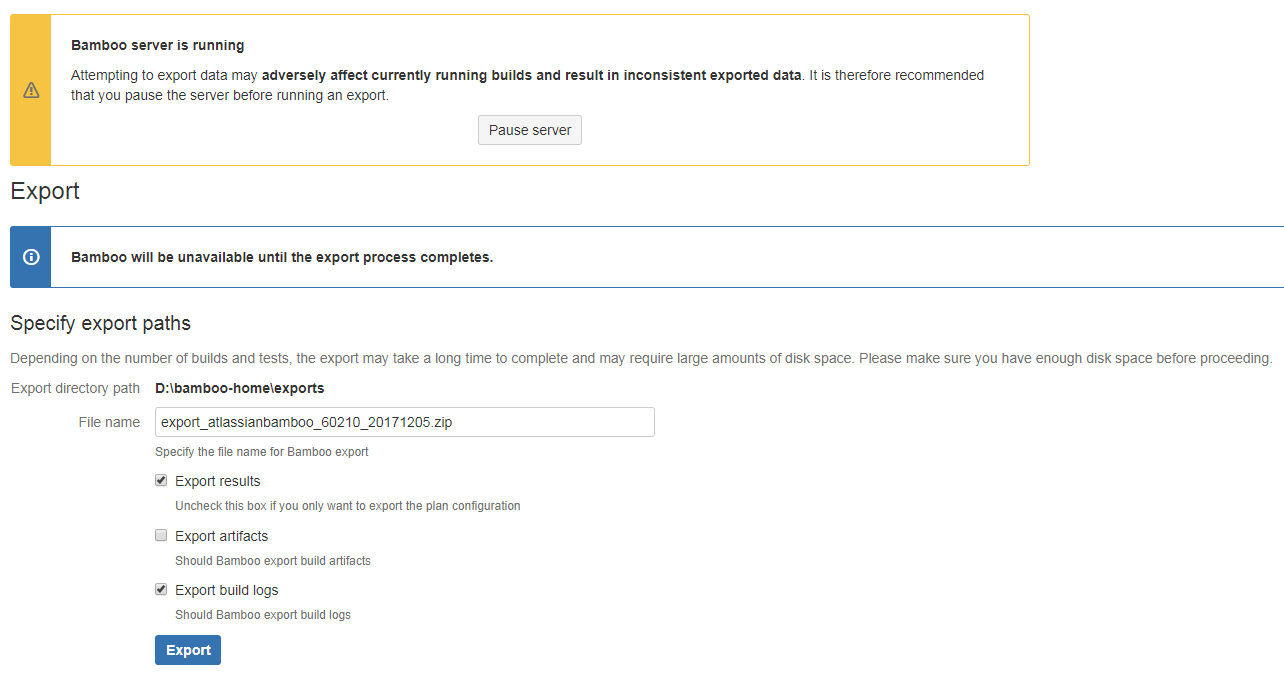
### Bamboo server backup

Bamboo server should be backed up periodically to have a rollback or repair option if the server gets corrupted (which has happened).

To back up all the current configurations, including users, groups, plans, agents etc.

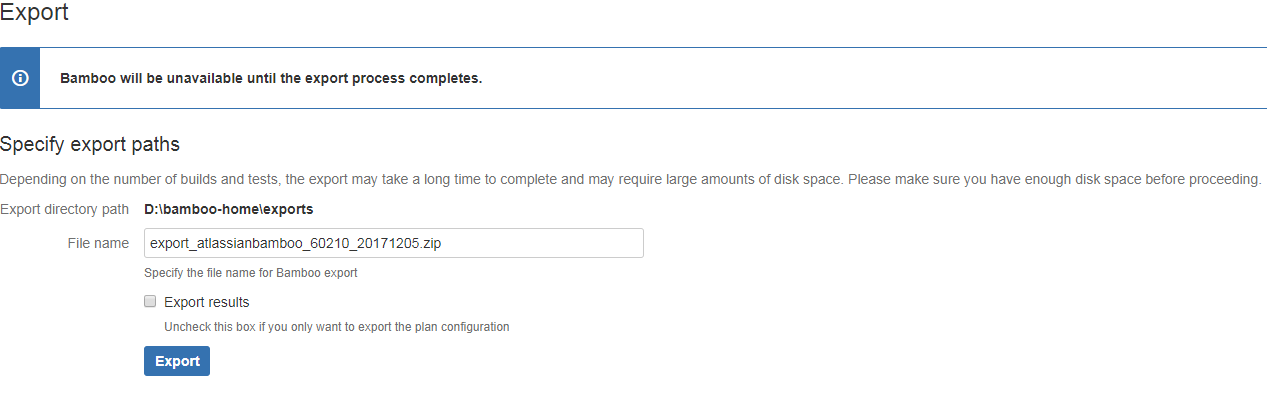
Navigate to the export page <http://172.28.114.109:8085/admin/export.action>

Fig. 14 — Export screen



Uncheck “Export results” checkbox

Click “Pause server” button



Click “Export”

Once the export completes, click “Resume server” in the right top corner

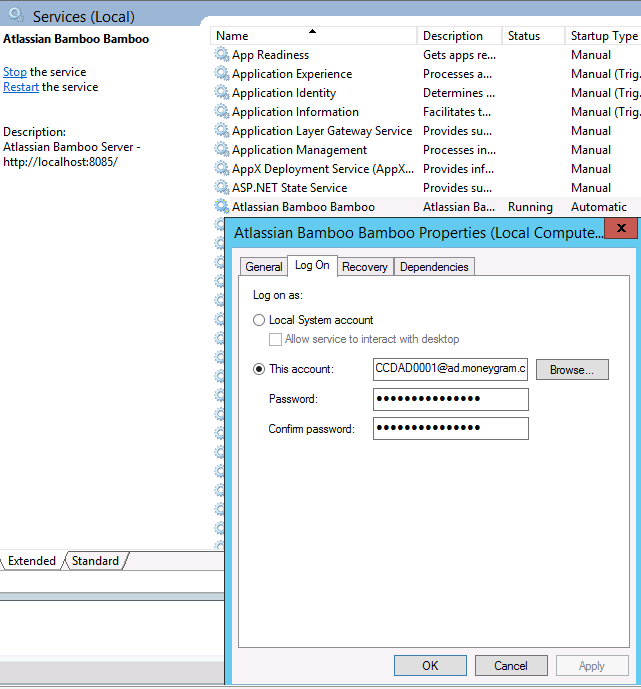
It is a good idea to copy the exported zip file off the server.

DONE.

### Bamboo server upgrade

1. Export current server configurations as described in [BAMBOO SERVER BACKUP](#_Bamboo_server_backup)
2. RDP and Log into the build server using CyberArc
3. Navigate to <https://www.atlassian.com/software/bamboo/download>
4. Click “Download” to download the installer
5. Open Services and stop “Atlassian Bamboo Bamboo” service
6. Start the downloaded installer
7. Once the install runs, re-install the service
8. Go to D:\Bamboo
9. Run “InstallAsService.bat”
10. Right-click on the service, select Properties in the context menu.
    1. Select Log on tab
    2. Select “This account” radio button
    3. Use [**CCDAD0001@ad.moneygram.com**](mailto:CCDAD0001@ad.moneygram.com) and **fg5A$D9poft6JM8mi9** for ID and PWD and Confirm PWD
    4. Click OK
11. Start “Atlassian Bamboo Bamboo” service
12. DONE

Fig. 15 — Adding service credentials to Atlassian Bamboo service



## Dev Server configuration, maintenance, administration, and management

Dev server is a virtual machine allocated to AgentWorks dev team for hosting various deployment environments.

Server: QMNACMS8104

ID: AD\CCQAD0011

PWD: get from FireCall, for instructions see <https://trello.com/c/GJ8TXTvr>

All IIS sites should be deployed under the share for IIS directories — [\\QMNACMS8104\AgentWorks](file:///\\QMNACMS8104\AgentWorks)

All logs should be written to the share for logs — [\\QMNACMS8104\AgentWorks\_logs](file:///\\QMNACMS8104\AgentWorks_logs)

### IIS Hosting directory structure

The root directory ([\\QMNACMS8104\AgentWorks](file:///\\QMNACMS8104\AgentWorks)) has 2 subdirectories

* Api — root directory for all Web API IIS environments
* StaticContent — root directory for all Static Content IIS environments

### Dev server logging directory structure

The root directory ([\\QMNACMS8104\AgentWorks\_logs\Api](file:///\\QMNACMS8104\AgentWorks_logs\Api)) contains subdirectories named exactly the same as the deploy environments whose logs they contain.

Make sure to clean up those directories periodically by deleting old log files.

### Dev server environments

Environments are documented on Trello <https://trello.com/c/Itv13inT>

Make sure to keep this card up-to-date.

At the time of the writing, these were the environments:

\*\*Dev server web sites for the \*master\* branch\*\*

\*\*Static Content\*\*

- aw\_sc\_master\_dev - <http://172.28.111.76>

- aw\_sc\_master\_qa - <http://172.28.111.76:8191>

- aw\_sc\_master\_fakeauth - <http://172.28.111.76:8192>

- aw\_sc\_master\_apidocs - <http://172.28.111.76:8195>

- aw\_sc\_master\_coverage - <http://172.28.111.76:8196>

\*\*Web API\*\*

- aw\_api\_master\_dev - <http://172.28.111.76:8090>

- aw\_api\_master\_fakeauth - <http://172.28.111.76:8092>

- aw\_api\_master\_qa - <http://172.28.111.76:8091>

\*\*Dev server web sites for the \*release\* branch\*\*

\*\*Static Content\*\*

aw\_sc\_rel\_dev - <http://172.28.111.76:8201>

\*\*Web API\*\*

aw\_api\_rel\_dev - <http://172.28.111.76:8093>

### Dev server configuration

Dev server has to have all the run-time dependencies installed, e.g. .NET 4.7, IIS URL Re-write.

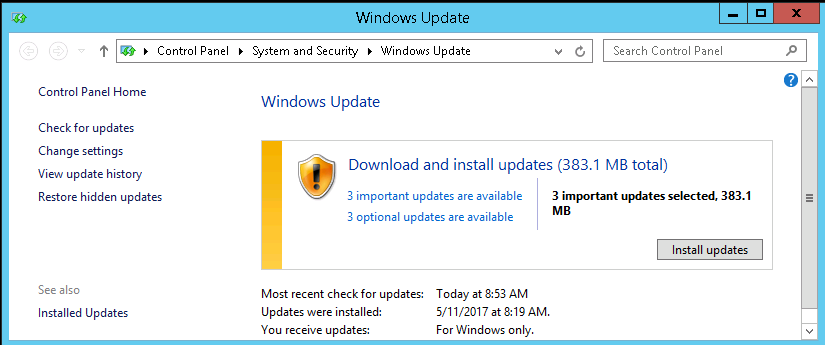
Dev server IIS node root directory must be exposed as a share and have all rights — read/write/delete — granted to the user Bamboo service is using. If write/read/delete rights are not granted, Bamboo deploys will not be able to run.

### Dev server maintenance

To update the server

1. Log into Dev server using CyberArc
2. Open File Explorer
3. Navigate to “Control Panel\System and Security\Windows Update”

Fig. 16 — Windows Update window



1. Install all important and optional updates
2. Reboot the server as instructed
3. DONE

IMPORTANT — Repeat the same process for the build server.

## Git code versioning

AgentWorks uses Git for code versioning.

At the time of this writing, Git is at version 2.15.1.2.

It is important for all developers and for Bamboo server to have git installed with the same settings.

### Git install

Git installer is hosted at <https://git-scm.com/>

Check the site periodically to monitor for version upgrades. Install new versions on all machines as they become available.

Make sure everyone on the team and both Dev and Build servers have Git installed with these configurations.

Fig. 17.1 — Adjusting your PATH environment screen. Select “Use Git from the Windows Command Prompt”

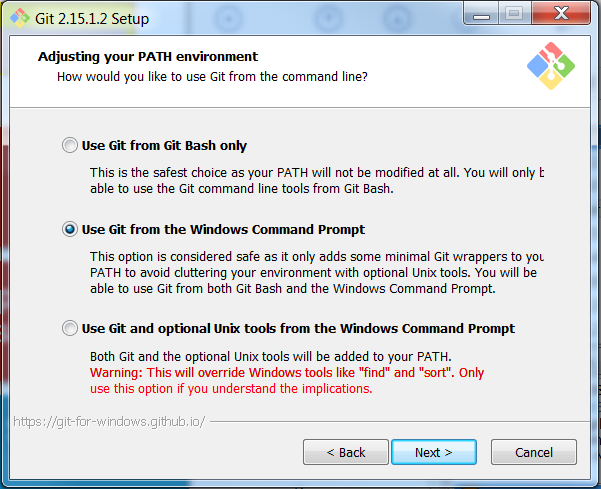


Fig. 17.2 — Choosing HTTPS transport backend screen. Select “Use OpenSSL library”



Fig. 17.3 — Configuring the line ending conversions screen. Select “Checkout as-is, commit as-is”

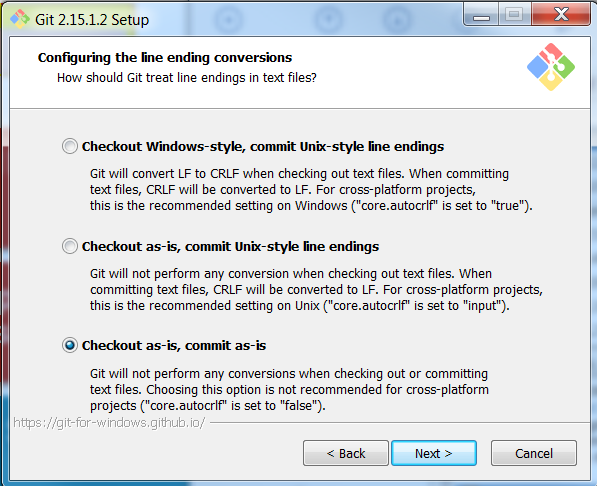


Fig. 17.4 — Configuring the terminal emulator to use with Git Bash screen. Select “Use MinTTY (the default terminal of MSYS2)”

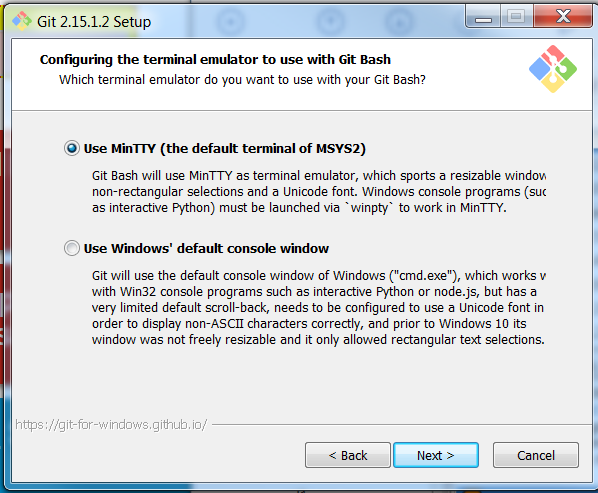
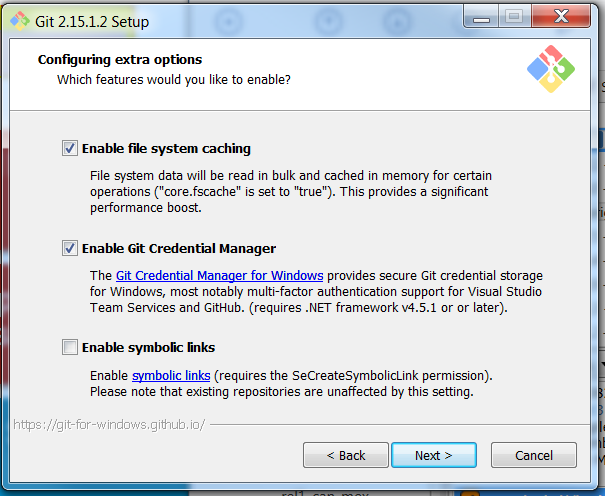


Fig. 17.5 — Configuring extra options screen. Check “Enable file system caching” and “Enable Git Credential Manager”



Click through to install with these options.

### Git Client

The teams’ default Git UI client is SourceTree.

SourceTree can be downloaded from <https://www.sourcetreeapp.com/>

Once it is installed, open Tools->Options

On General tab, make sure to check “Always display full console output”

On Git tab, make sure that Use System Git is selected

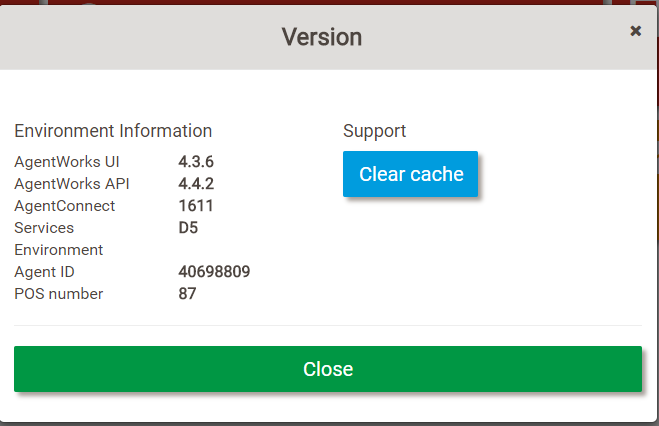
### Git code versioning, branching and merging

#### Versioning

Here are the guidelines that have been used for code versioning.

* AgentWorks Web API and Static Content code bases are versioned separately
* Semantic Versioning formatting is used for versioning, while the version increment rules are custom. Read the standard <https://semver.org/>
* Major version (**1**.2.3) is incremented once a release
* Minor version (1.**2**.3) is incremented once a sprint
* Patch version (1.2.**3**) is incremented with each commit
* Commit messages should be formatted as follows “v[VERSION] — [COMMIT\_MESSAGE]. Code review by [CODE\_REVIEWER\_NAME]”. Example “v4.4.2 — fixed DT4 role authorization for Profile query. Code review by Max H.”
* For merges, the format of the commit message is “v[VERSION] — MERGE FOR — [COMMIT\_MESSAGE\_OF\_ORIGINAL COMMIT]. Code review by [CODE\_REVIEWER\_NAME]” Example “v4.4.0 — MERGE FOR — HOTFIX — Asynchronously logging using log4net. Code review by Dmitri M”
* Both API and SC versions can be viewed in the application by clicking on the Version link

Fig. 18 — Version popup



#### Branching

When the development reaches a stable point before the release, a release candidate branch must be created.

To create a Git branch

* Make sure you are on the branch FROM WHICH you want to create the new branch. This most likely will be master.
* Make sure you have no local code changes. If you do, commit them or revert them.
* Pull latest
* Make sure everything builds and runs
* Run command: git checkout -b [NEW\_BRANCH\_NAME] where NEW\_BRANCH\_NAME is the branch name that you want to create
* Run command: git push origin [NEW\_BRANCH\_NAME]
* The new branch is now available to the team

#### Release candidate branch naming

When you create a branch for an upcoming release, the name should include the release number and major functionality being released.

Example: rel3-dt4tran

This name indicates that it is for release 3 and the release delivers DT4 transactional.

#### Release candidate versioning

When Release branches are created, the initial commit should update the version to the next release’s pre-release format.

Example: 4.0.0-rc.1.0.1

This number means that this branch was cut from the master branch where the MAJOR version is 3.

4.0.0 part of the version is the next production release version.

-rc.1.0.1 part means this is a release candidate version 1, commit 1

When the UAT and regression testing is complete, update the version to 4.0.0. From that point on, increment PATCH number only.

Update master branch to 4.1.0 and going forward apply the standard increment rules (PATCH for each commit, MINOR for each sprint, e.g. 4.1.1, 4.2.0).

The HOTFIXes will only increment the PATCH in the release version (e.g. 4.0.1)

All release and hotfix branch commits must be merged into the master branch.

## BitBucket repositories management and administration

AgentWorks team home page — <https://bitbucket.org/AgentWorks/>

AgentWorks project home page — <https://bitbucket.org/account/user/AgentWorks/projects/AW>

Fig. 19 — Repositories under AgentWorks team

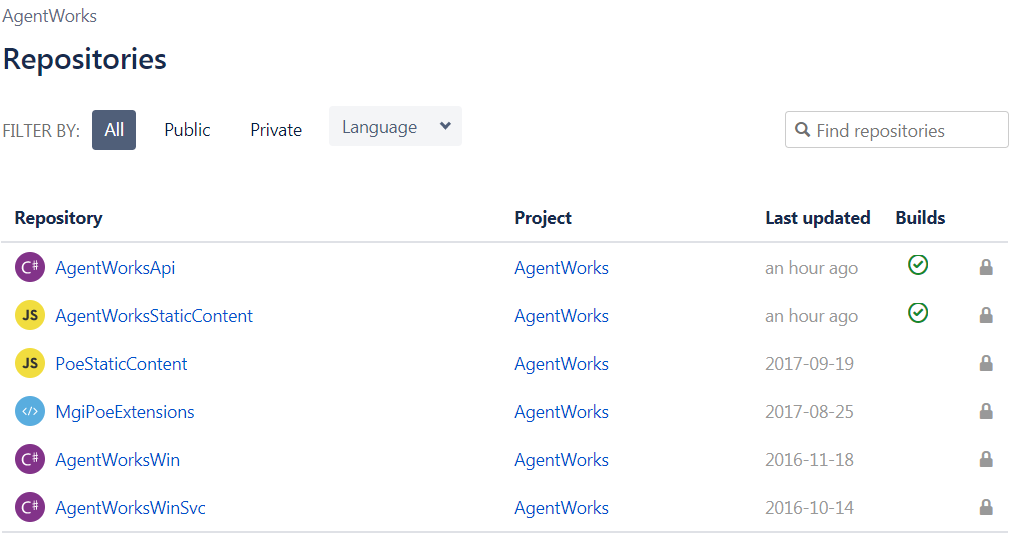


Fig. 20 — AgentWorks project settings <https://bitbucket.org/account/user/AgentWorks/projects/AW/settings>



### BitBucket repository Overview page

Fig. 21.1 — AgentWorksApi repository Overview page <https://bitbucket.org/AgentWorks/agentworksapi>

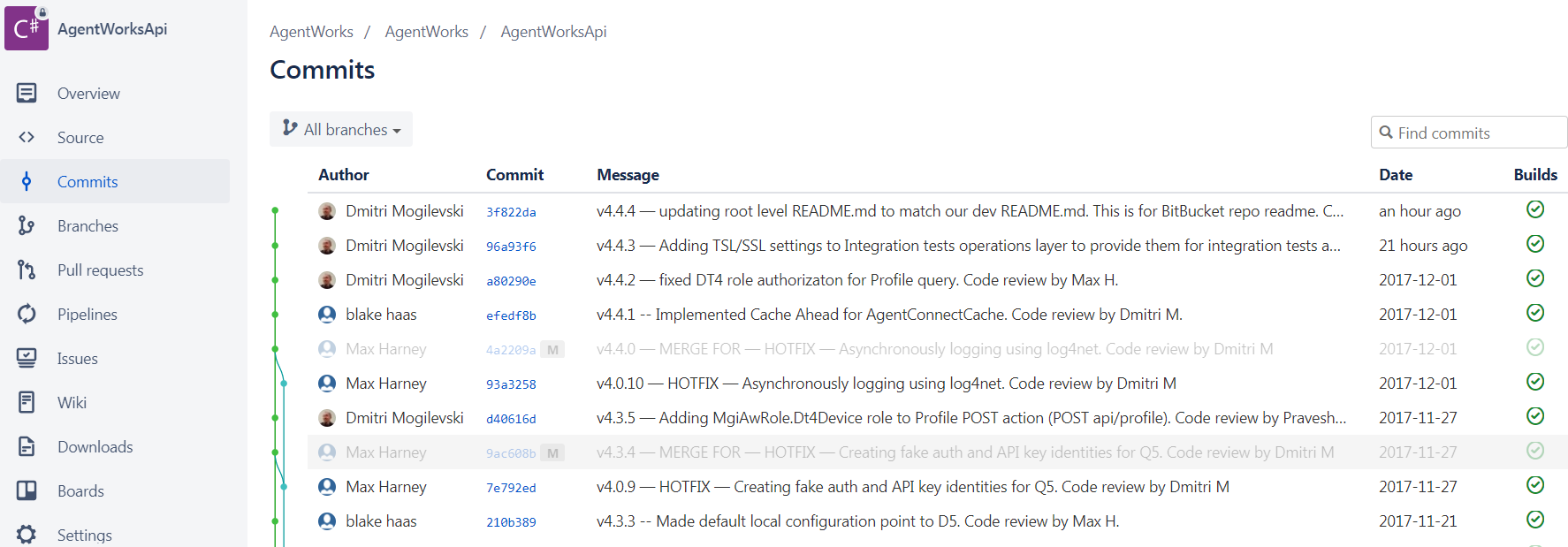


The overview page gives developers easy access to the clone URL and the top level repository menu which is on the left.

If the root directory of the repository contains README.md file, BitBucket will display it on the overview page.

### BitBucket repository Commits page

Fig. 21.2 — AgentWorksApi repository Commit page <https://bitbucket.org/AgentWorks/agentworksapi/commits/all>



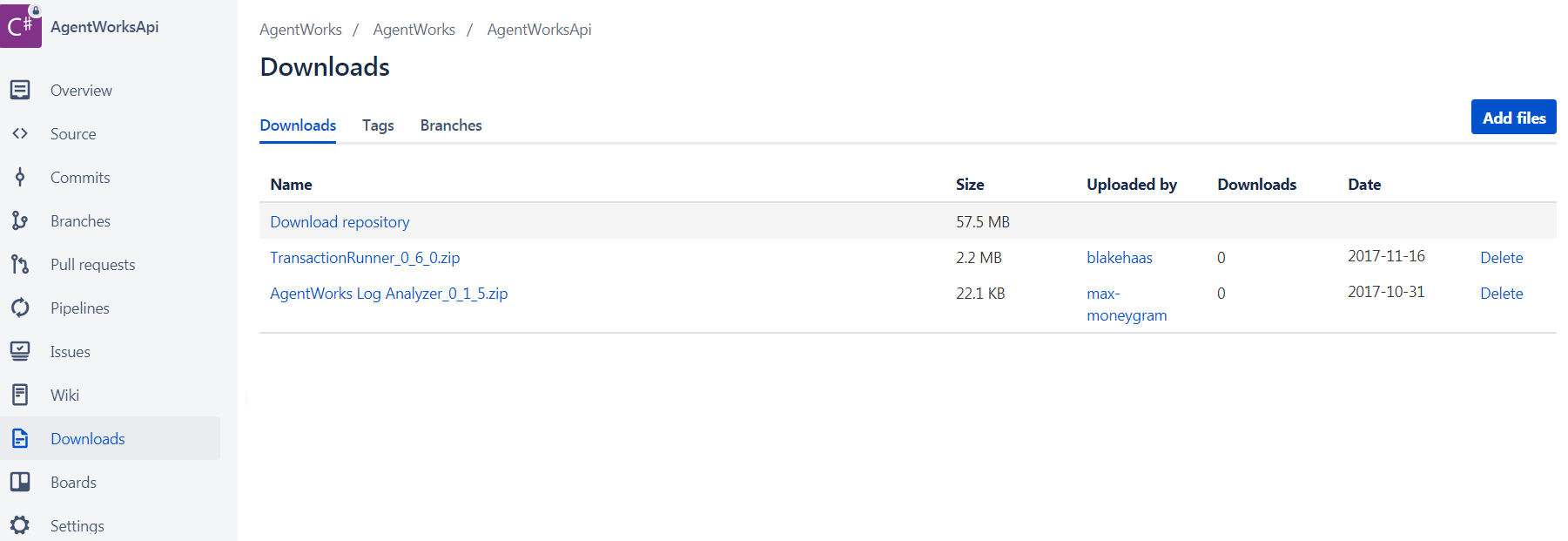
This is a list of commits that exposes commit hashes, messages, and build statuses. Build status is a link that will navigate user to the Bamboo page for the build.

Use the Branches dropdown to filter commits by branch, if needed.

Find commits filter allows to search for commit messages that contain a specific string.

### BitBucket repository Downloads page

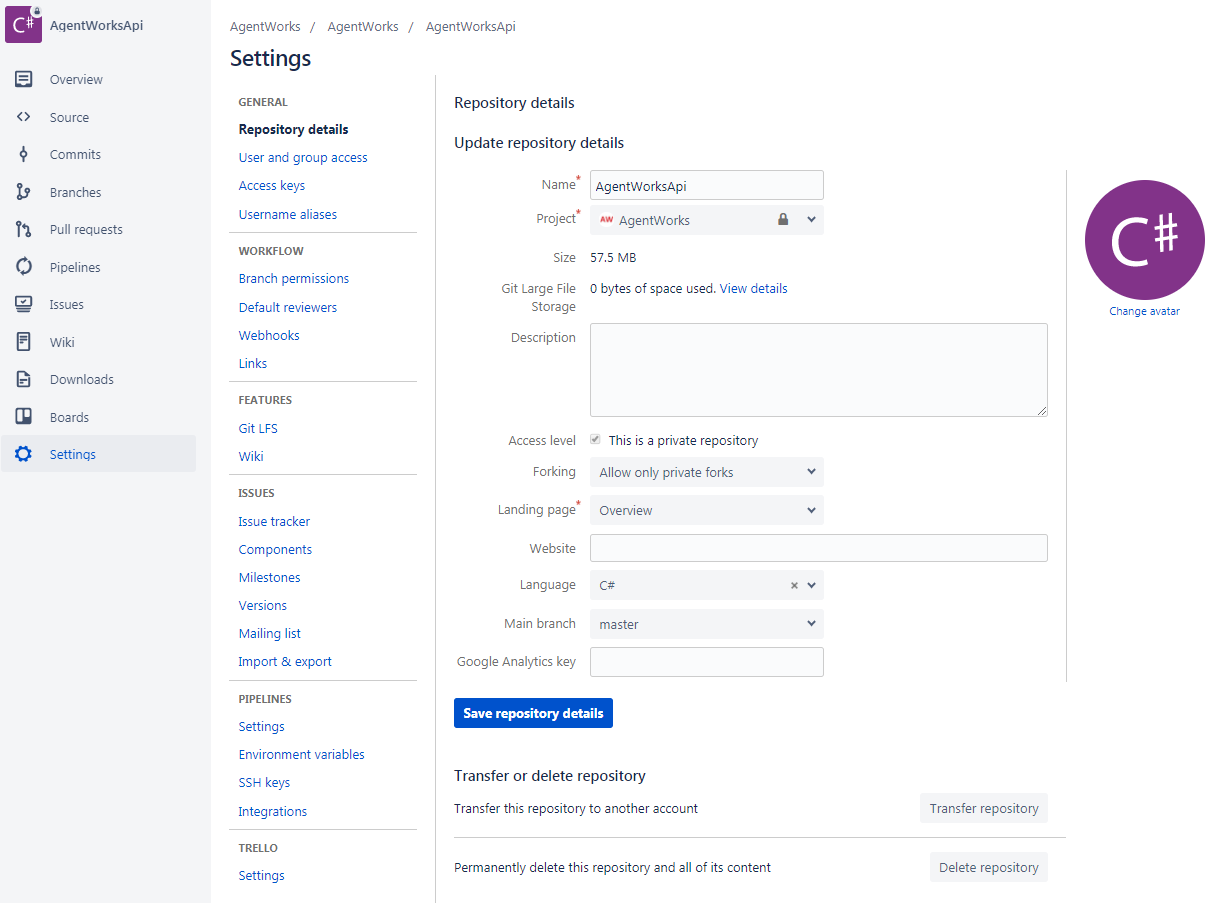
Fig. 21.3 — AgentWorksApi repository Downloads page <https://bitbucket.org/AgentWorks/agentworksapi/downloads/>



This page is where files can be added for sharing with other teams (Product, Performance testing etc.) To get access to it, users must have Observer privileges to the repository.

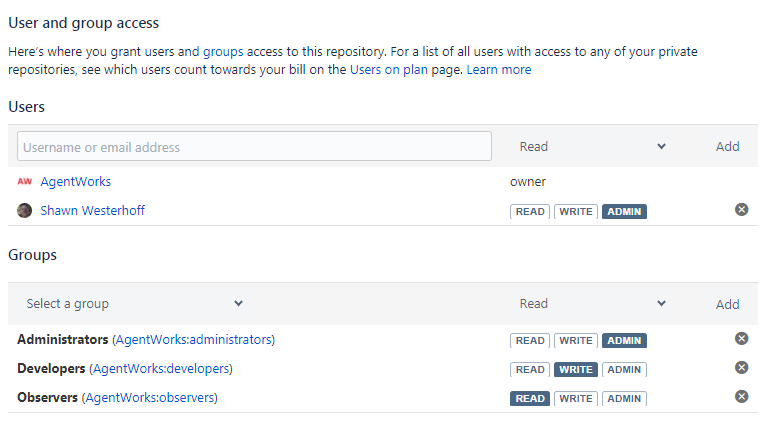
### BitBucket repository Settings page

Fig. 21.4 — BitBucket repository Settings page <https://bitbucket.org/AgentWorks/agentworksapi/admin>



This is the place to configure access rights, groups, branch permissions, webhooks etc.

Fig. 21.4.1 — BitBucket repository Settings — User and group access page <https://bitbucket.org/AgentWorks/agentworksapi/admin/access>



All access should be controlled by Group membership.

Administrators have all rights.

Developers can clone and commit, create branches etc but cannot admin the repository.

Observers can clone but cannot commit or admin.

# Development

## Product Owner collaboration

Meet with Product Owner at least once a sprint to fine-tune the understanding of product vision and roadmap.

Try to keep it informal and brief. Ideally, have a plan that you can collaborate on with Product.

Review and demo very frequently to give Product a chance to correct a design, add business rules, adjust priorities

**Product is the only source of truth** about design, business rules, — everything.

Drive unblocking the team, take ownership of issues

## Story readiness assessment

For external dependencies, it is important to understand them completely and have a good validation strategy. Example — if we need to integrate with a card reader, what make/model do we support? Do we have test cards that cover all supported kinds for testing? If not, who will be responsible for getting them for us?

Does the story fit into the overall strategy of Product?

Is now the right time to pick the story up?

## Code quality enforcement processes

Consistency of architectural design

Enforce code placement in the correct functional and application domain, in other words, make sure that the new code is written in the correct component and code base.

Keep the code style and readability consistent

Variables must be explicitly named and scoped to the least level possible

Functions must be consistently and descriptively named

A component should ideally do one thing only – single responsibility

If something is done more than once, make it a sharable functionality – DRY

If external requests violate our architectural design, bring it up and explain the long-term cost

## Code reviews

Code reviews must be performed for each commit and merge.

Code reviews should be used to review design decision and enforce code quality principles.

After the review, it is good to the new feature with Product.

## Nuget updates

Always keep the app patched with the latest – that is the goal

Be mindful of release boundaries – during the release, switching branches and merging becomes more difficult if the nuget versions are not the same

Update at least once a sprint

If a nuget package introduces an API change, create a story for the update. Fix what the update breaks.

For major dependencies, like Unity and log4net, create a façade so that ideally only 1 project is dependent on the nuget package. The goal is to decouple the layers of the application and to reduce the impact of package’s breaking API changes

## NPM Updates

Same principles as the Nuget updates apply.

## Authentication and authorization in the Web API

### Summary

Web API is built on the OWIN standard (<http://owin.org/>).

OWIN kicks off the hosting process by executing an app-defined Startup class. That class must have a public Configuration method that takes a parameter of type IAppBuilder.

This mechanism allows developers to wire up middleware components to process different aspects of HTTP request/response pipeline.

Authentication and authorization components are wired up into the OWIN pipeline and responsible for

* Validating the security token with the provider
* Building a Principal and adding all roles and claims to it
* Authorizing access to controller endpoints based on roles and claims
* Supplying Principal data for integrations with external services (e.g. AC)

Web API exposes multiple authentication providers and, as a result, supports multiple access mechanisms. Enabling and disabling of providers is done through appSettings.config properties.

For all financial integrations in production, AgentWorks integrates with OpenAM and OpenIDM.

OpenAm and OpenIDM integration supports 3 Principal types

* User – the actor is a human system user, authenticated by providing credentials to OpenAM which then issues an access token for the user session
* Registered Device — the actor is a known device that has gone through the registration step
* Unregistered Device — the actor is a known device that has not gone through the registration step

Fake Auth provider was created to facilitate development and ease of identity switching. It emulates OpenAM user authentication, but the identity is validated and constructed by Web API without OpenAM.

API Key provider was created to facilitate Web API stand-alone testing from Swagger or HTTP request tools. It emulates OpenAM user authentication, but the identity is validated and constructed by Web API without OpenAM.

Support Key is similar to API Key, but it was created for production to facilitate a limited non-financial endpoint access by external systems. It supports a very small subset of claims.

Authentication and Authorization providers are wired up in Startup.cs

// Configure Auth providers

var authDependencyManager = (IAuthDependencyManager) configuration.DependencyResolver.GetService(typeof(IAuthDependencyManager));

AuthProviderManager.ConfigureAuthProviders(app, authDependencyManager);

AuthProviderManager.ConfigureAuthProviders method goes through all known providers and performs these steps

* Check if the provider is allowed in a given environment by looking at the property in the appSettings.config that governs this provider
* If it is allowed, register it with the AuthTracker
* Wire in the Authentication middleware class into OWIN pipeline
* Wire in the Principal middleware class into OWIN pipeline

On every HTTP request, each allowed authentication middleware component will be invoked and go through the steps to determine whether this request has the required inputs for its principal. If so, it validates the input, constructs the principal. If all registered authentication providers fail to validate input and construct a principal, an Unauthorized response will be returned.

### OpenAM/OpenIDM

OpenAM is the SSO provider for AgentWorks/Agent Portal

It issues access tokens to the static content, which then sends them in Authentication HTTP header as Bearer tokens.

On the Web API side, the bearer token is

Sent to the OpenAM tokeninfo endpoint for validation

Send to the OpenAM userinfo endpoint for user claims

The claims obtained from OpenAM are then used to construct a principal.

#### User

User principal is composed from claims from OpenAM, PartnerService, and AC

ClaimsInventoryHelper class verifies whether we receive all required claims from each source.

Required OpenAM user claims are listed in the ExpectedOpenAmClaims property.

Required PartnerService user claims are listed in the ExpectedPartnerServiceClaims property.

Required Agent Profile user claims are listed in the ExpectedAgentProfileClaims property.

#### Registered device

Registered device principal is composed from claims from OpenAM and PartnerService

ClaimsInventoryHelper class verifies whether we receive all required claims from each source.

Required OpenAM user claims are listed in the ExpectedCamsDeviceClaims property.

Required PartnerService user claims are listed in the ExpectedPartnerServiceClaims property.

#### Unregistered device

### Fake auth

### API key

### Support key

### Localization and Internationalization in Web API

### Localization and Internationalization in Static Content

# Tools List

## Development tools

Web API is developed in C#, .NET 4.7, Visual Studio 2017. MSBuild 15 is required for the build server. .NET 4.7 is required on all IIS servers that host Web API.

Static Content is developed in JavaScripAngularJs 1.6.7, Bootstrap, ui-router, lodash. The recommended development tool is WebStorm, however developers are also using Visual Studio Code.

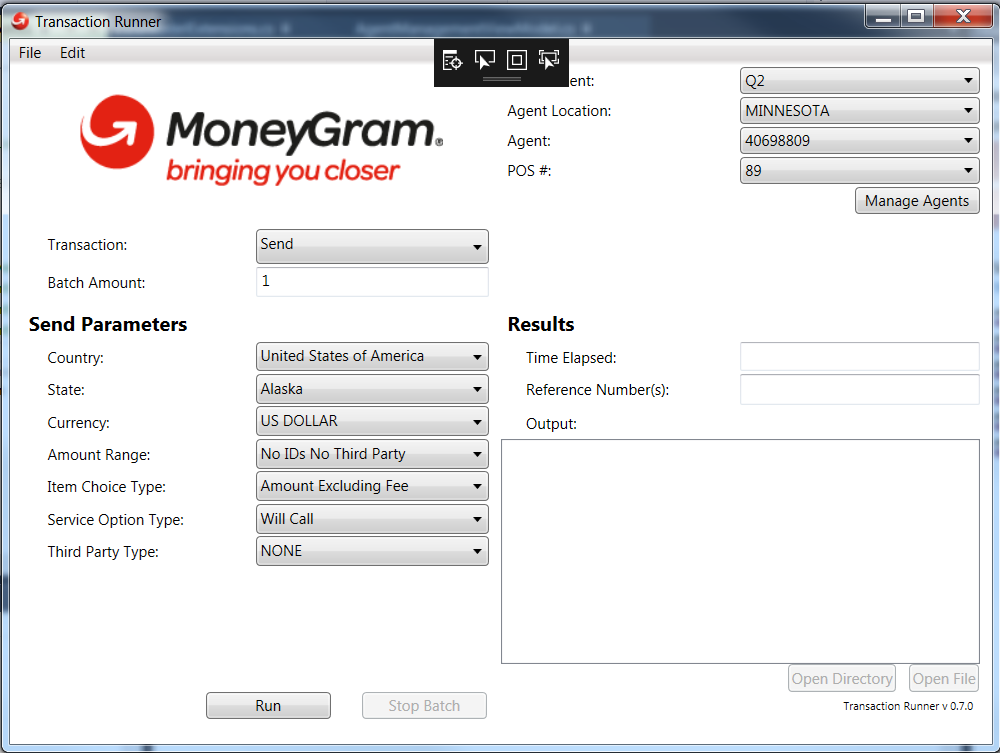
IIS must be installed on developer machines for running local sites.

## Application ecosystem tools

All source code for the tools is under the Web API repository, under /agentworksapi/Tools directory.

### TransactionRunner

TransactionRunner is a UI client that runs the same code as Web API integration tests. It uses the operations that integrate with AC functionality. The difference between integration tests and TransactionRunner is that it exposes the “arrange” part of tests as UI screens and provides identity configuration and batching capabilities.



Available transaction types can be run by selecting the type from the Transaction dropdown.

Batch Amount field controls how many transactions will be run as a batch — this is really useful for creating test data, staging multiple transactions, performance testing etc.

### AcUpgradeTool

This tool was built to generate data classes, data mapping extensions, JavaScript data objects etc from AC WSDL. Refer to the documentation in the source /AgentWorksApi/Tools/AcUpgradeTool/AcUpgradeTool/Documentation

### AwLogAnalyzer

Source code located is under /AgentWorksApi/Tools/AwLogAnalyzer

This tool will read Web API logs and produce analysis logs.

### LangProcessor

LangProcessor is a tool that processes source language files into the JSON files that the language API uses for localization.

The source code is located under /AgentWorksApi/Tools/LangProcessor.

Language source files are under /AgentWorksApi/Languages/Applications/AgentWorks/source

When you update any source file, update LangSourceMetadata.json with the new timestamp and the LangProcessor will process the source files and produce json files in the dist directory — /AgentWorksApi/Languages/Applications/AgentWorks/dist. The updated dist directory files must be checked in with the rest of the changes, these are the actual files that will be packaged and used by the Web API.

LangSourceMetadata.json contents

{

"en-US": {

"Version": "**2017-12-14T21:48:59.272Z**"

},

"fr-CA": {

"Version": "**2017-12-14T21:48:59.272Z**"

},

"es-MX": {

"Version": "**2017-12-14T21:48:59.272Z**"

},

"de-DE": {

"Version": "**2017-12-14T21:48:59.272Z**"

}

}

## Swagger – API documentation and testing

Swagger is an API documentation specification and package.

The integration with AgentWorks Web API is in the AwApi.Documentation project.

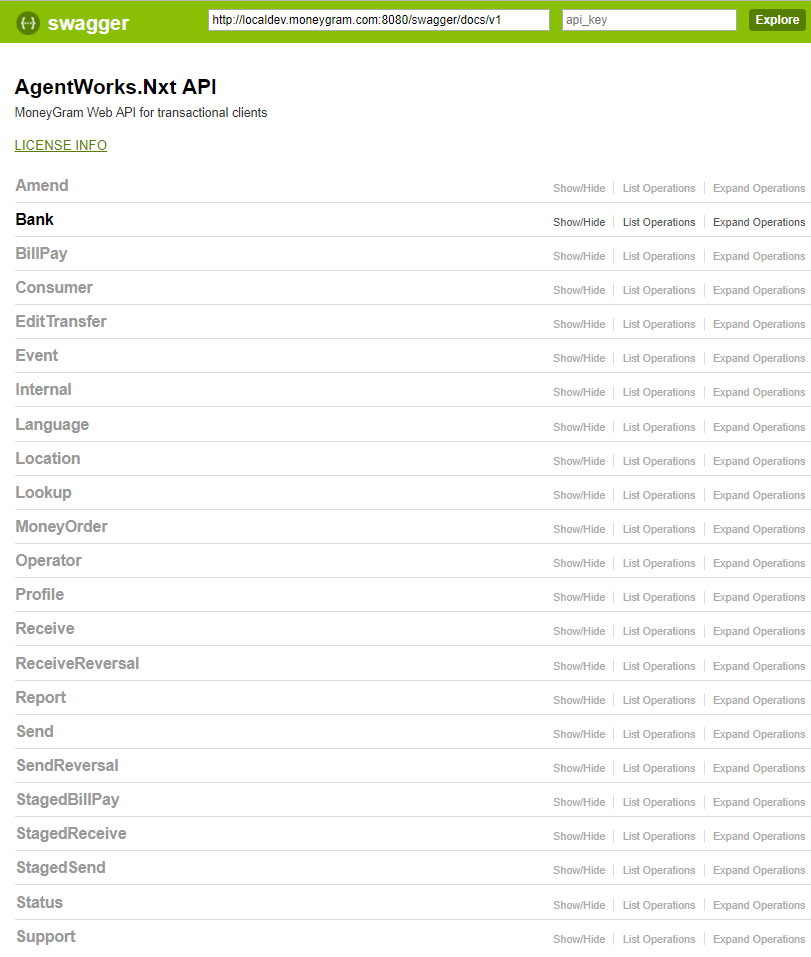
That project is dependent on Swashbuckle.Core Nuget package. This nuget package is an implementation of swagger.

The documentation availability is configured by environment in “ApiDocumentation” setting in appSettings.config.

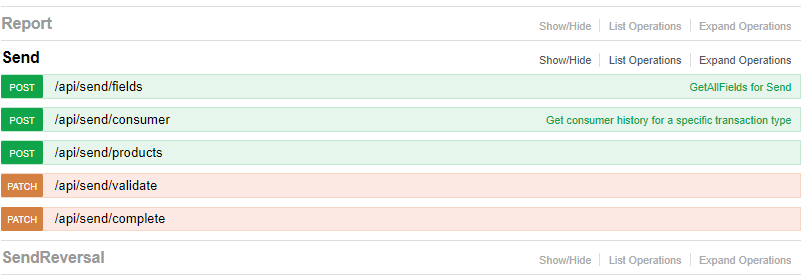
If the setting is set to true, the swagger API documentation is available at

[API URL]/swagger/ui/index#/ e.g. <http://localdev.moneygram.com:8080/swagger/ui/index#/>

Full Web API surface



Expanded Send controller endpoints

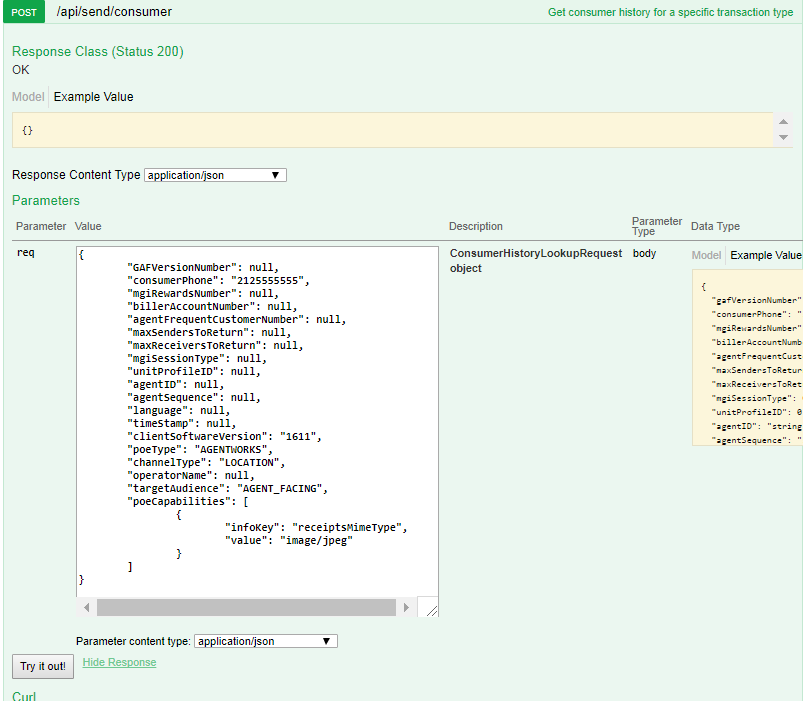


Executing POST /api/send/consumer in Swagger

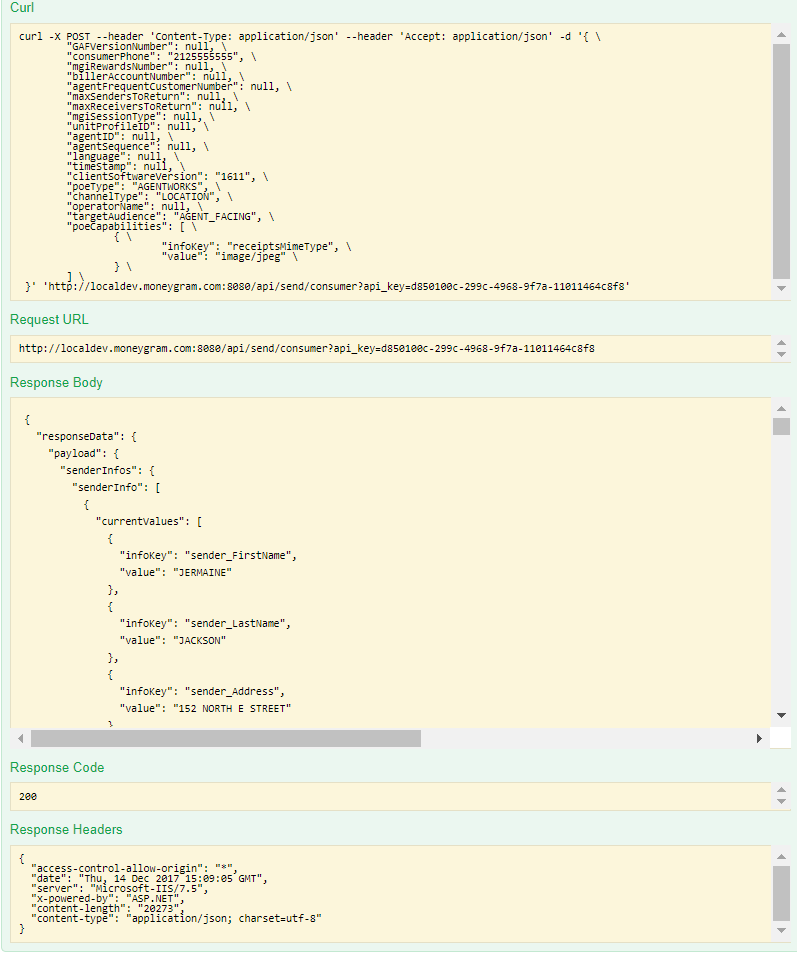
API Key that resolves to the identity this request will act as



Request populated



The full curl request, Request URL, and Response Body/Code/Headers



The entire flows may be orchestrated and executed through Swagger.

The API Keys for each environment are defined in apiKeys.json.

Add new identities to apiKeys.json if the existing ones are not sufficient for a given use case.

apiKeys.json contains identities that mimic User principals constructed from OpenAM/PartnerService/AgentProfile claims and roles.

Example of an API Key identity with corresponding claims and roles:

"d850100c-299c-4968-9f7a-11011464c8f8": {

"MgiAppAwMaxSendAmount": "3500",

"Sub": "userMN87",

"MgiAppAwMaxBillPayAmount": "1000",

"MgiAgentLocationId": "40698809",

"MgiUserLastLoginTS": "20170605140504.933",

"Given\_Name": "MN",

"Country": "USA",

"MgiDevicePosNumber": "87",

"MgiDeviceAgentId": "40698809",

"Updated\_At": "0",

"MgiMainOfficeId": "40698786",

"MgiAppAwMaxReceiveAmount": "3200",

"Email": "userMN87@moneygram.com",

"Name": "userMN87",

"Family\_Name": "POS87",

"Language": "en-US",

"MgiPosUnitProfileId": "3080004",

"MgiAppAwMaxSendCancellationAmount": "1000",

"MgiAppAwMaxReceiveCancellationAmount": "1000",

"MgiAppAwRoles": [

"APP - AgentWorks - Teller",

"APP - AgentWorks - Administrator"

],

"agentPassword": "456",

"agentName": "John Smith",

"storeName": "DELTAWORKS TEST #1",

"agentTimeZone": "GMT-6:00",

"agentAddress1": "1550 UTICA AVE S",

"agentAddress2": "",

"agentTelNo": "9525913600",

"primaryReceiptLanguage": "eng",

"secondaryReceiptLanguage": "spa",

"canSend": "true",

"canSendCompletion": "true",

"canAmend": "true",

"canCancel": "true",

"canRefund": "true",

"canFeeRefund": "true",

"canReceive": "true",

"canReceiveCompletion": "true",

"canReceiveReversal": "true",

"canBillPay": "true",

"canBillPayCompletion": "true",

"canLoadPrepaidCard": "false",

"canMoneyOrder": "false",

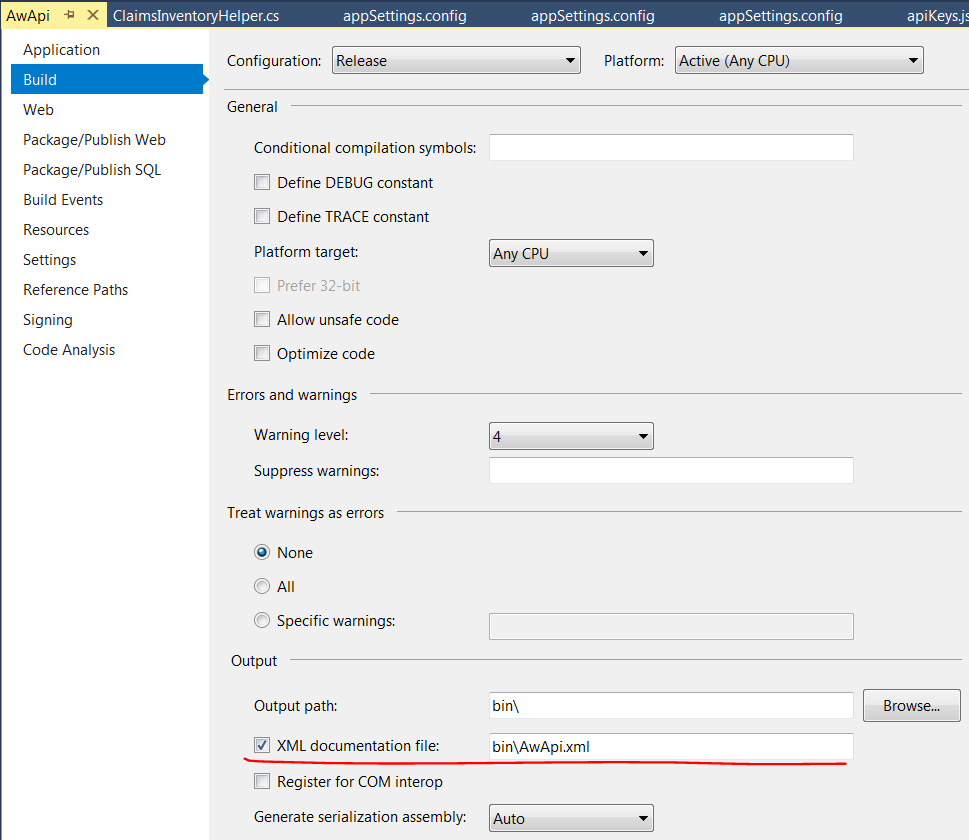
"canVendorPay": "false",

"canExpressPayment": "false",

"canUtilityBillPay": "true"

}

In order for swagger to build the documentation site, the build configuration of the API and ViewModel projects should output the XML documentation files.



## Trello – knowledge repository

The Trello team homepage is <https://trello.com/moneygramagentworks>

Sunita and Pravesh are admins.

The team is private, no one outside the team has access to the boards.

The main team board is <https://trello.com/b/BlyyJcLl/moneygram>

This is where the team keeps common instructions, setup details, environment details etc.

### Add new people to the team

Have them register at Trello.com and send you their user name or email address they used for registration.

Then navigate to the Members page — <https://trello.com/moneygramagentworks/members>

Click on “Add by Name or Email” on the left and paste in their user name/email.

## Slack – team collaboration and DevOps notifications

Slack is an instant message, media, and workflow management tool.

Our Slack team name is awm-dev (stands for AgentWorks MoneyGram development).

Manage the team settings at <https://awm-dev.slack.com/admin/settings>

Manage the team membership at <https://awm-dev.slack.com/admin>

Sunita is the team owner

Pravesh is an admin on the team

Custom integrations are managed at <https://awm-dev.slack.com/apps/manage/custom-integrations>

Currently, Slack is integrated with BitBucket and Bamboo. All DevOps related messages are being posted to the #devops channel.