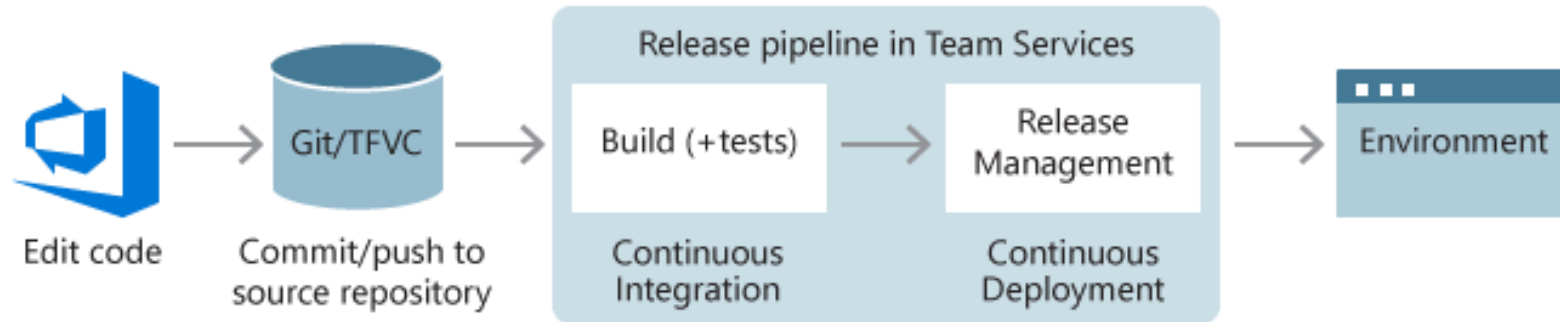




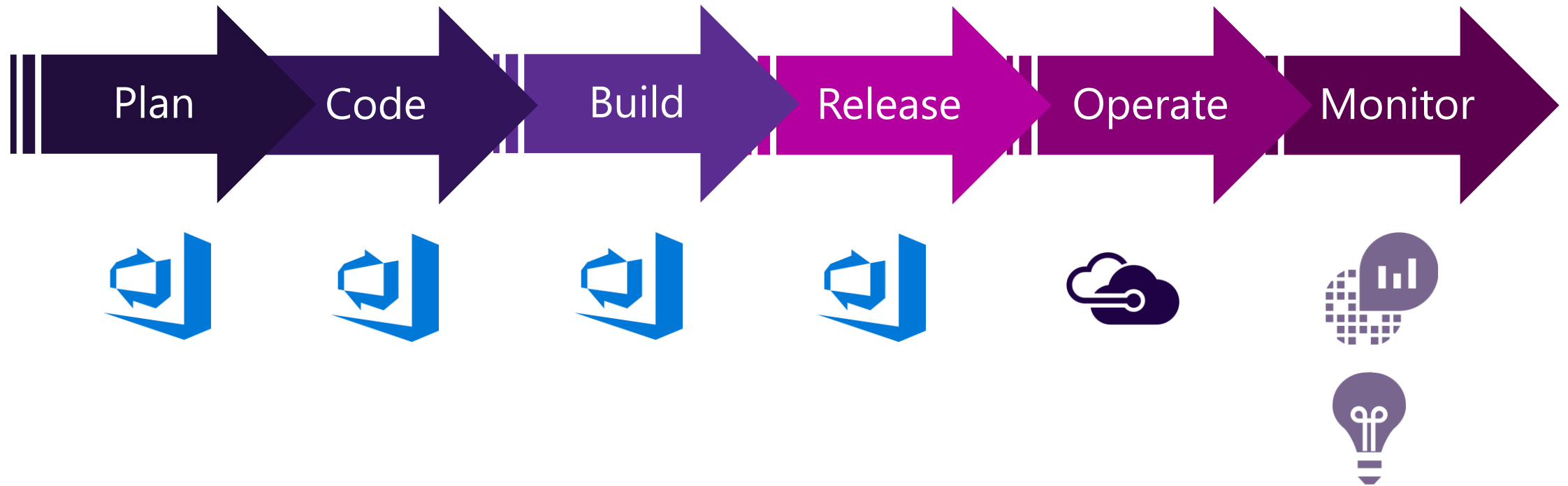
CI/CD

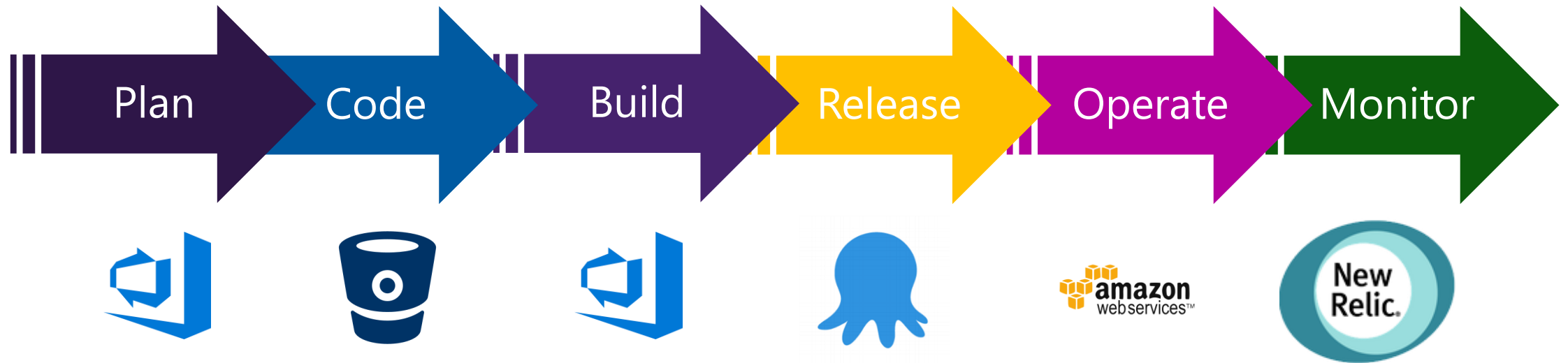
Crystal Tenn
Crystal.Tenn@microsoft.com

CI/CD Overview



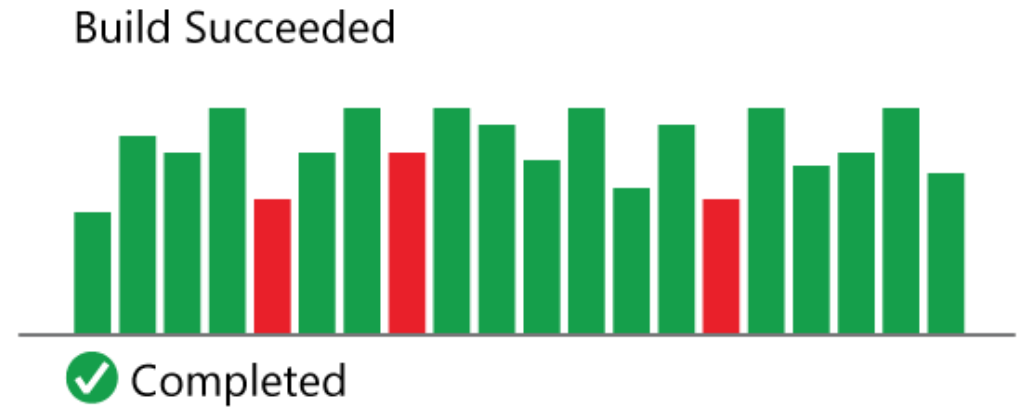
- CI means starting an automated build (and possibly running tests) whenever new code is committed to or checked
- CD means starting an automated deployment process whenever a new successful build is available.
- Together, CI and CD = live results fast





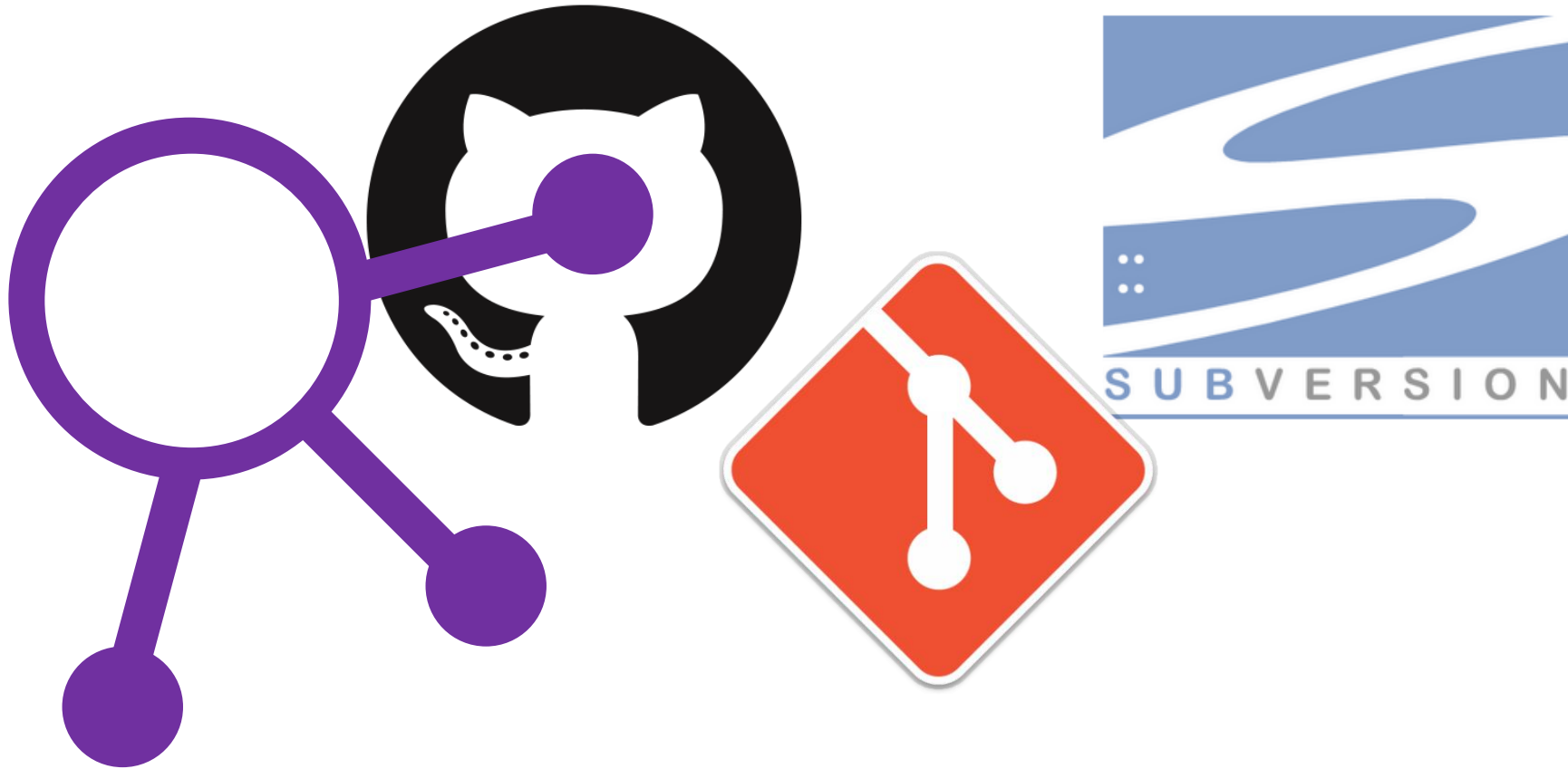
What is Continuous Integration?

Continuous Integration



- Continuous Integration (CI) is the practice of merging code and automatically running a build process including tests whenever a developer commits code changes.

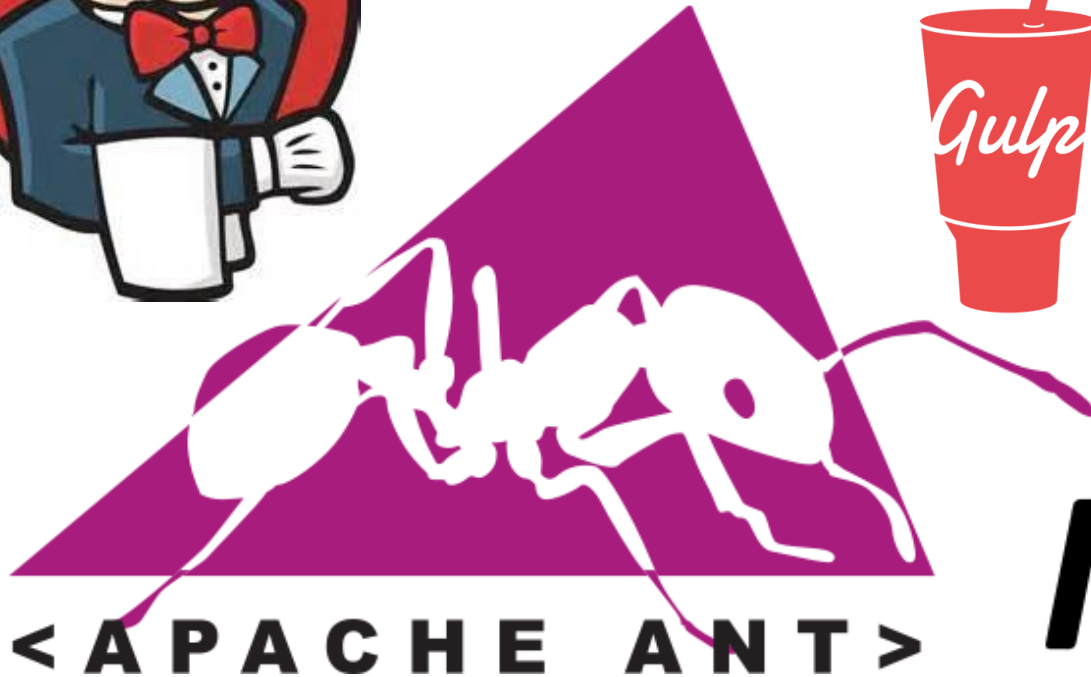
Out of the box: Source Control Integration



Out of the box: Build Integration

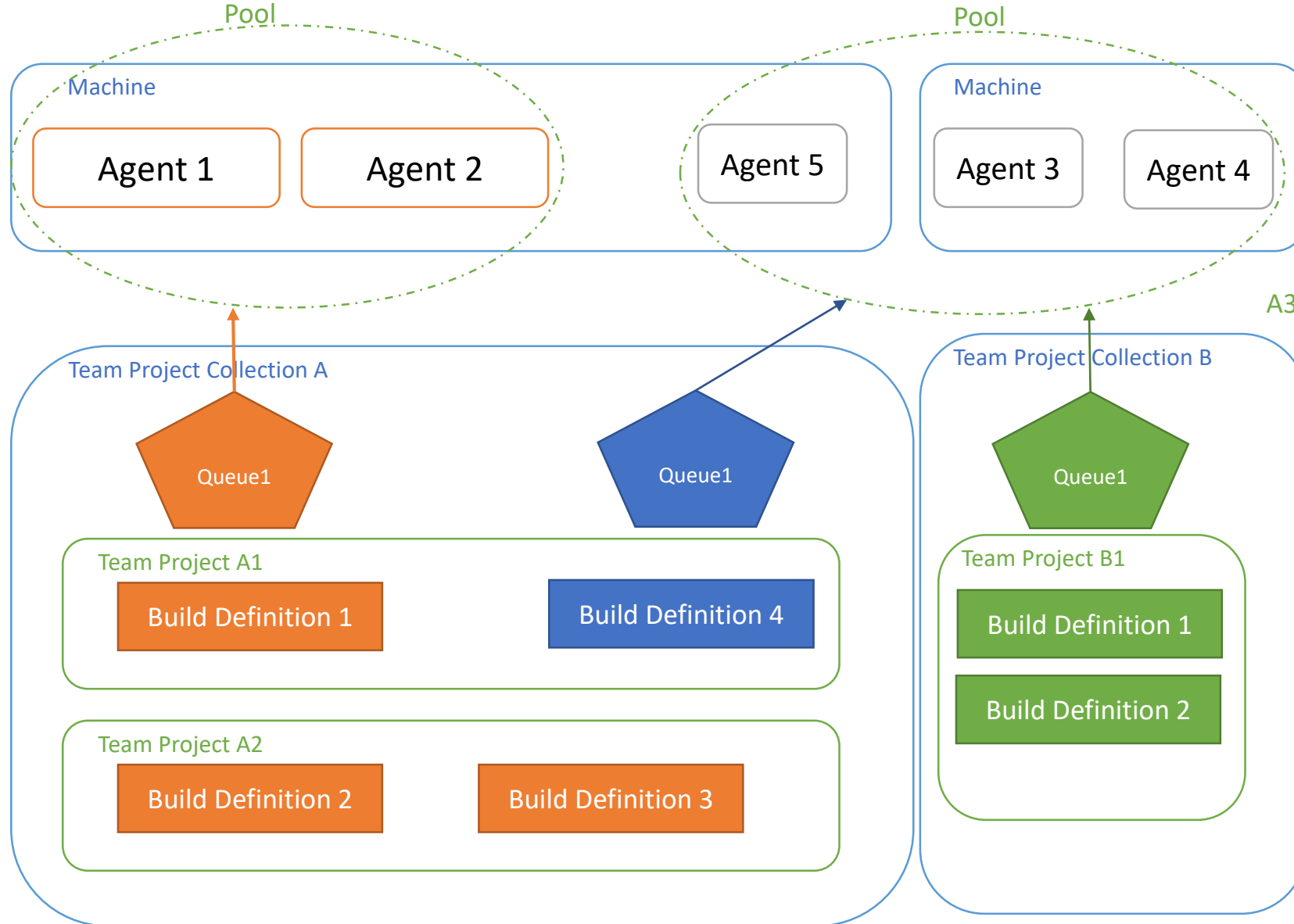


gradle



MavenTM

Build – Architecture



Q1: I need to build apps on a server

A1: You need an Agent

Q2: We might have many build / Teams

A2: Add more agents, scale them via a Pool

Q3: How will my projects access these pools

A3: They will use a queue to talk to the pool

A3: Your build definition will specify the Queue it will use

A3: Many build definitions can use the same queue

Q4: Anything else I should know

A4: You may have many build servers

A4: Containing other agents

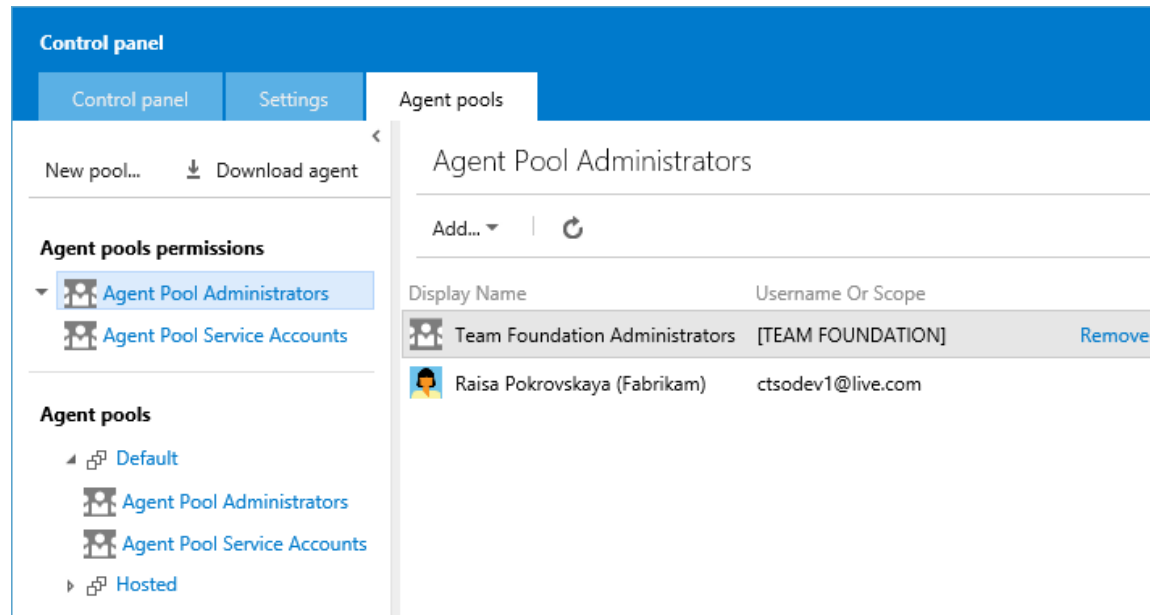
A4: Pools can span multiple machines

A4: You need a pool even for one agent

A4: Collections can share pools

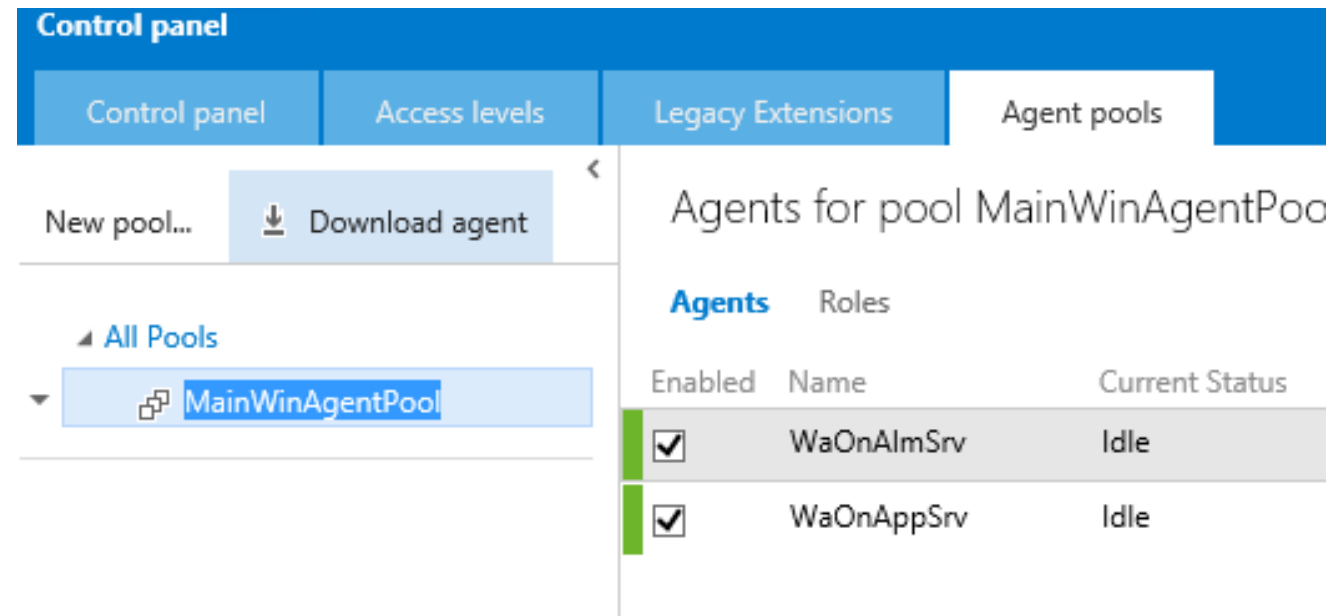
Build – Why Pools?

- Organize and define permission boundaries.
- Scalability by leveraging multiple agents.



The screenshot shows the 'Agent pools' control panel. The left sidebar contains 'Agent pools permissions' with 'Agent Pool Administrators' and 'Agent Pool Service Accounts', and 'Agent pools' with 'Default', 'Agent Pool Administrators', 'Agent Pool Service Accounts', and 'Hosted'. The main area is titled 'Agent Pool Administrators' and shows a table of administrators.

Display Name	Username Or Scope	
Team Foundation Administrators	[TEAM FOUNDATION]	Remove
Raisa Pokrovskaya (Fabrikam)	ctsodev1@live.com	



The screenshot shows the 'Agent pools' control panel with the 'MainWinAgentPool' selected. The left sidebar shows 'All Pools' with 'MainWinAgentPool' highlighted. The main area is titled 'Agents for pool MainWinAgentPoo' and shows a table of agents.

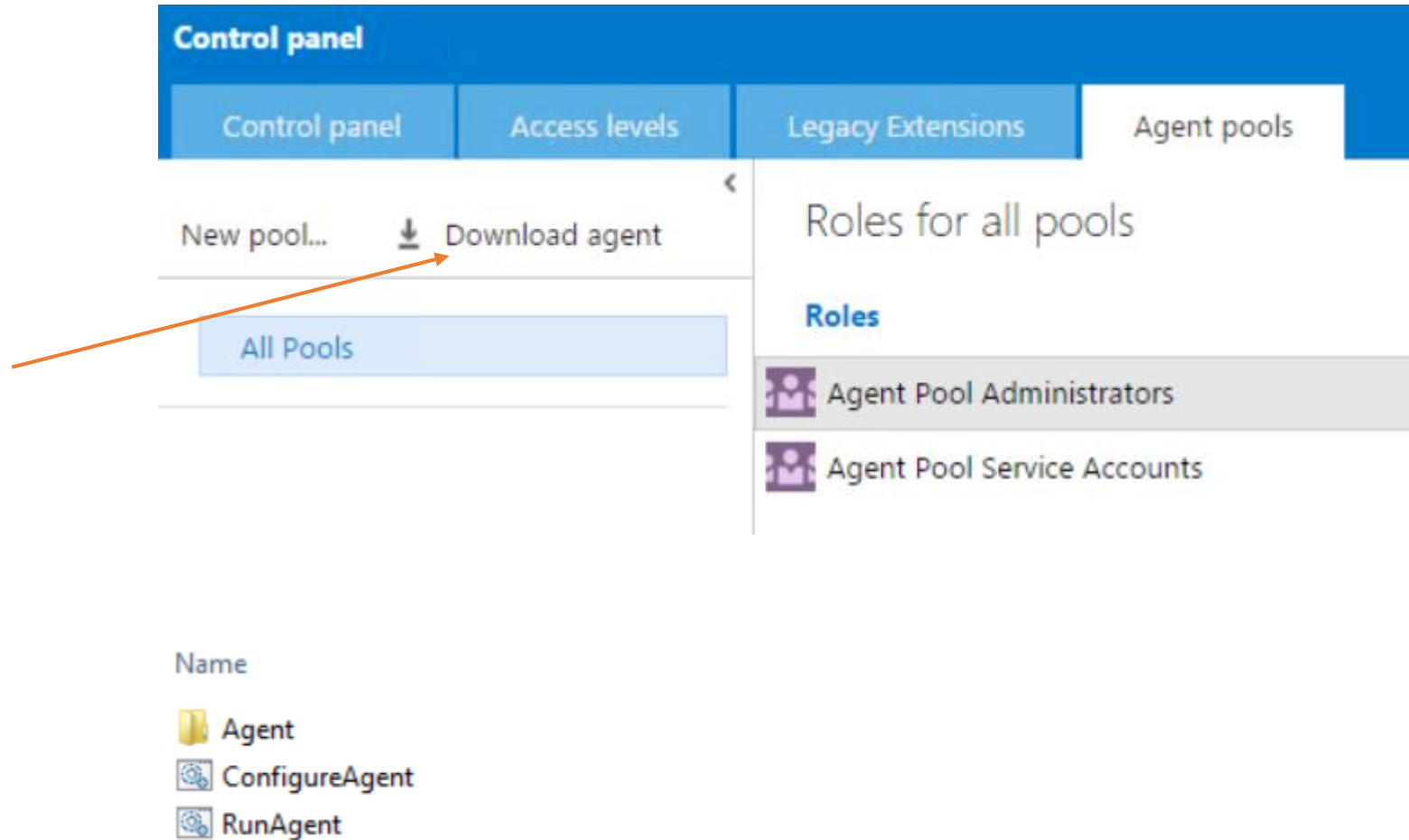
Enabled	Name	Current Status
<input checked="" type="checkbox"/>	WaOnAlmSrv	Idle
<input checked="" type="checkbox"/>	WaOnAppSrv	Idle

Build – Why Queues

- An agent queue provides access to a pool of agents.
- Build definitions specify queues.
- Scoped to a collection.
- Permissions can be set at many levels:
 - To grant global permissions, select All Queues, and then to grant permission:
 - To manage all queues, select Agent Queue Administrators.
 - To create queues, select Agent Queue Creators.
 - To enable people to create definitions that use any queue, add them to Agent Queue Users
 - To grant narrower permissions, select a queue and add people to either the administrators role or the users role

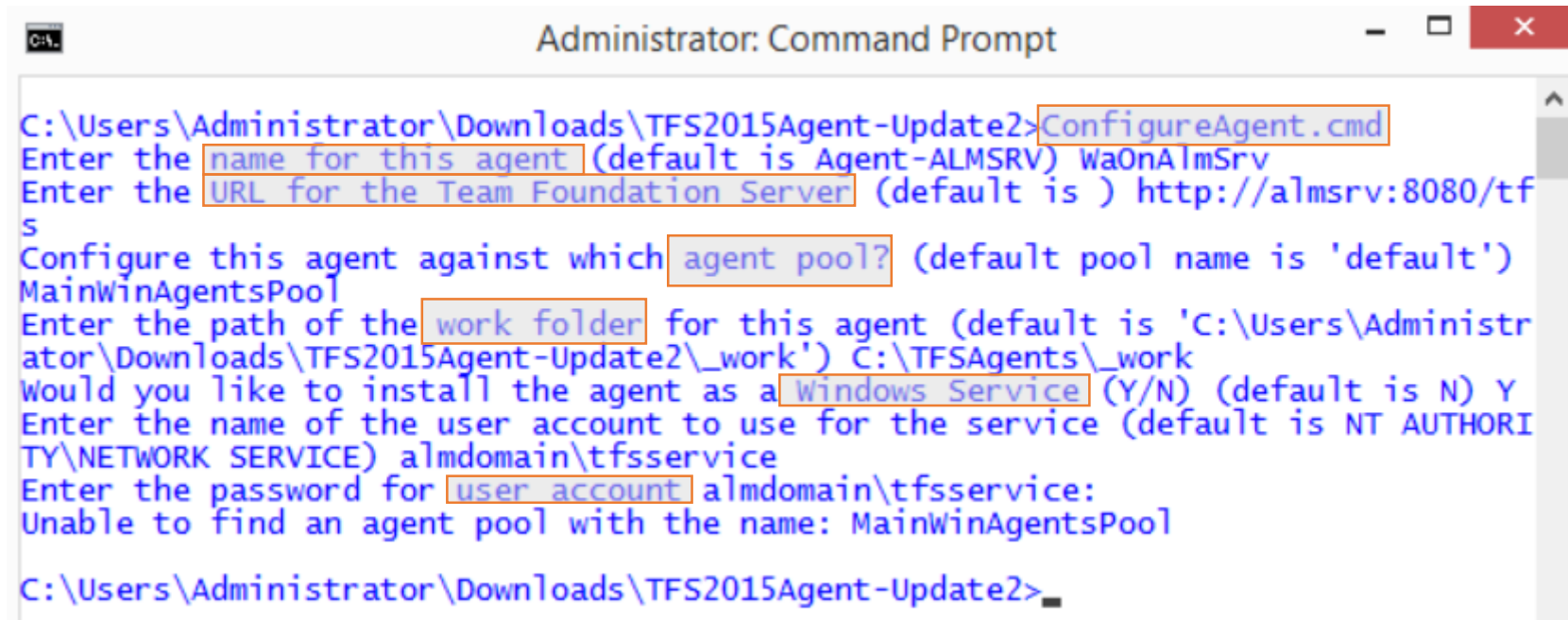
TFS Build – Deploy an Agent

1- Download the agent (installer) from your TFS/VSTS instance



TFS – Deploy an Agent

2- Run the ConfigureAgent on the Build Server.



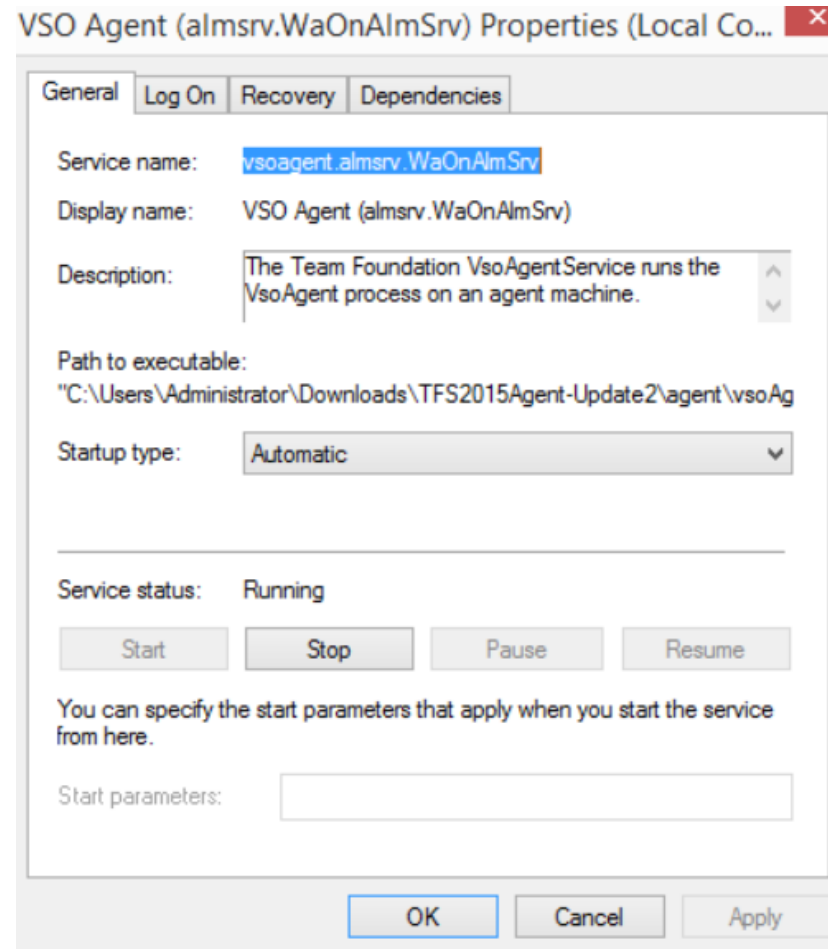
```
Administrator: Command Prompt

C:\Users\Administrator\Downloads\TFS2015Agent-Update2>ConfigureAgent.cmd
Enter the name for this agent (default is Agent-ALMSRV) waOnAlmSrv
Enter the URL for the Team Foundation Server (default is ) http://almsrv:8080/tfs
Configure this agent against which agent pool? (default pool name is 'default')
MainWinAgentsPool
Enter the path of the work folder for this agent (default is 'C:\Users\Administrator\Downloads\TFS2015Agent-Update2\_work') C:\TFSAgents\_work
Would you like to install the agent as a Windows Service (Y/N) (default is N) Y
Enter the name of the user account to use for the service (default is NT AUTHORITY\NETWORK SERVICE) almdomain\tfsservice
Enter the password for user account almdomain\tfsservice:
Unable to find an agent pool with the name: MainWinAgentsPool

C:\Users\Administrator\Downloads\TFS2015Agent-Update2>
```

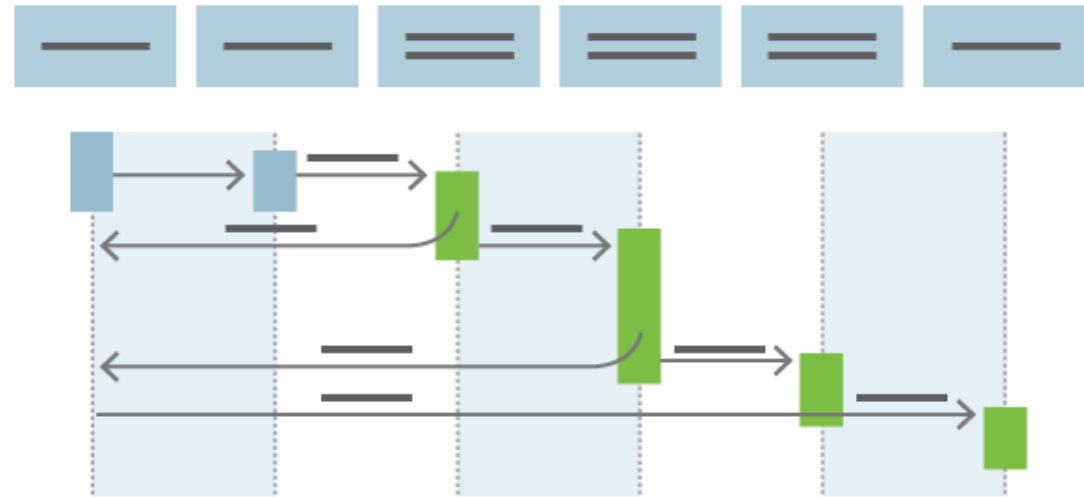
TFS 2015 Build – Deploy an Agent

3- Verify agent service is running



What is Continuous Delivery?

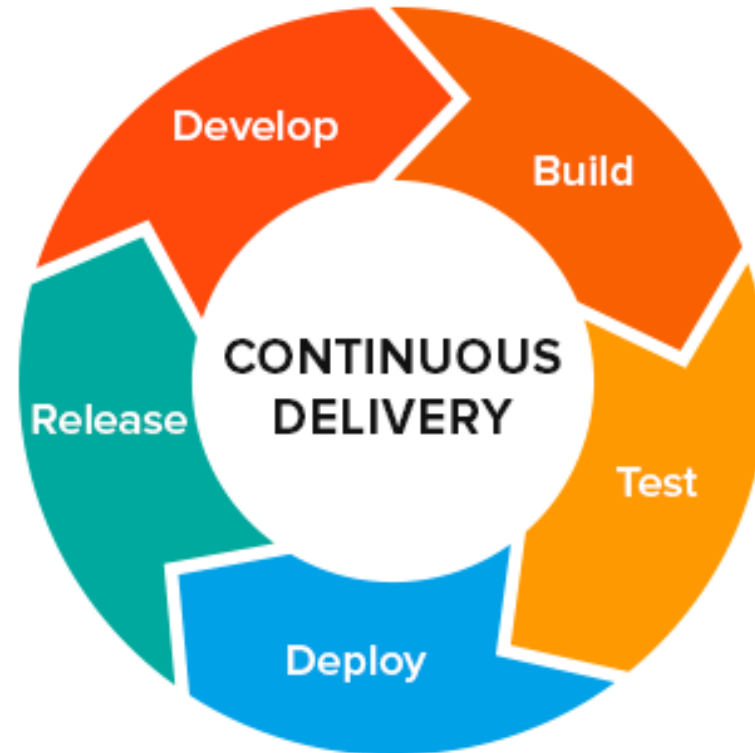
Continuous Delivery



- Continuous Delivery (CD) is the process to build, test, configure and deploy from a build to a production environment.

Continuous Delivery

- Without Continuous Delivery, software release cycles were previously a bottleneck for application and operation teams.
- Keep production fresh by achieving the shortest path from the availability of new code to deployment.



Out of the box: Release Integration





Visual Studio

Visual Studio Team Services

Visual Studio Code

Subscriptions

[Build your own](#)

integration



48 Results


Showing: All categories ▾

Hosted On: Any ▾

Price: Any ▾

Sort By: Relevance ▾



IBM Integration Bus
Microsoft  3

Manage and Deploy to IBM Integration Bus

★★★★★

FREE



Team Services Git Inte
BigBrassBand

View your Team Services Git commits inside of JIRA Issues and Projects + Code Review

★★★★★

FREE



Jenkins Integration
Microsoft

Jenkins is an award-winning, cross-platform, continuous integration and continuous

★★★★★

FREE



GitHub Integration
Microsoft

Continuous integration and deployment for your .NET, Java, Node.js, Android, iOS,

★★★★★

FREE



Azuqua Integration
Microsoft

Connect the services your teams and customers use, automating business

★★★★★

FREE



Slack Integration
Microsoft

Team communication for the 21st century integrated with your software development

★★★★★

FREE



Campfire Integration
Microsoft

Stay up-to-date with your Visual Studio Team Services projects from your team's

★★★★★

FREE




Flowdock Integration
Microsoft

Chat and inbox for Visual Studio Team Services teams. One place to talk and stay up-

★★★★★

FREE



Docker Integration
Microsoft  2.5K

Build, push, run or deploy Docker images and multi-container Docker applications.

★★★★★

FREE



Hubot Integration
Microsoft

Hubot is your helpful, personal assistant for Visual Studio Team Services.

★★★★★

FREE



UserVoice Integration
Microsoft

UserVoice is the the Product Management Platform for building better products.

★★★★★

FREE



Trello Integration
Microsoft

Assign work to teams using Trello, by creating cards on boards when events happen

★★★★★

FREE



Microsoft Teams Integration
Microsoft

Private chat, and file sharing that is fully integrated with your development

★★★★★

FREE



SCVMM Integration
Microsoft  279

Connect to a SCVMM server and perform actions on Virtual Machine(s) like

★★★★★

FREE



MyGet Integration
MyGet

Securely create, host and share NuGet, symbols, NPM, Bower and VSIX artifacts.

★★★★★

FREE



Zendesk Integration
Microsoft

Unite your customer support and development teams. Quickly create or link work

★★★★★

FREE



Zapier Integration
Microsoft

Connect and automate tasks between Visual Studio Team Services and hundreds of

★★★★★

FREE



Atlassian Bamboo Integration
Microsoft

Bamboo is a continuous integration and delivery server from Atlassian.

★★★★★

FREE



Office 365 Integration
Microsoft



JFrog Artifactory Integration
JFrog  207



Octopus Deploy Integration
Octopus Deploy  2.6K



Microsoft Teams Integration
Microsoft



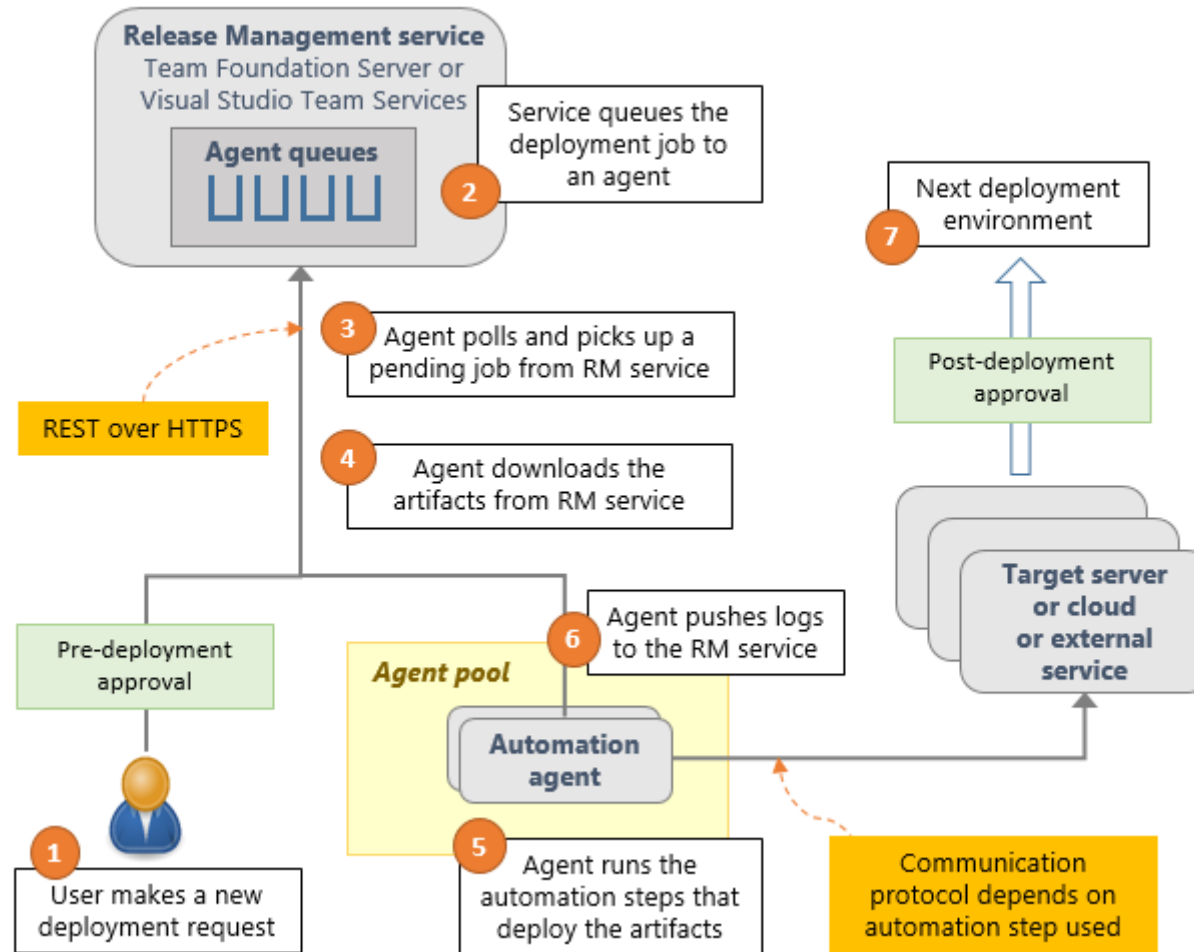
LaunchDarkly Integration
LaunchDarkly  225



Azure Service Bus Integration
Microsoft



How does it work




Release Definitions




A release definition is one of the fundamental concepts in Release Management. It defines:

- The types and sources of **artifacts** that make up new releases.
- The collection of **environments** in which the artifacts can be deployed.
- The automation **tasks** that can be executed in each environment

Release Definitions – Manual Scheduling

Definition: FabrikamDev  | [Releases](#)

Environments | Artifacts | Configuration | **Triggers** | General | History

 |  Save |  Release ▼





☒ **Manual**
Create a new release manually using the "Release" action.

☐ **Continuous Deployment**
Creates a new release every time a new artifact version is available.

☐ **Scheduled**
Create a new release at a specified time.

Environment triggers

Environment triggers specify when and how a deployment will be triggered on an environment.

Environment	Trigger	
Dev	Manual	
QA	Manual	
Production	Automated: after release creation	

No releases are initiated automatically when a new build of the source artifacts occurs. All releases for this release definition **must be created manually** by choosing the Release icon in a release definition or from a build summary

Release Definitions – “Continuous” Deployment

☐ **Manual**

Create a new release manually using the "Release" action.

☒ **Continuous Deployment**

Creates a new release every time a new artifact version is available.

Set trigger on artifact source

FabrikamDev



☐ **Scheduled**

Create a new release at a specified time.

A new release is **created automatically** when Release Management detects new built artifacts are available.

Release Definitions – Scheduled

☐ **Manual**

Create a new release manually using the "Release" action.

☐ **Continuous Deployment**

Creates a new release every time a new artifact version is available.

☒ **Scheduled**

Create a new release at a specified time.

24h time: : (UTC) Coordinated Universal Time

Su ☐ M ☒ Tu ☒ W ☒ Th ☒ F ☒ Sa ☐

A new release is created based on a **schedule** you specify.

Release Definitions – Continuous Deployment

Environment triggers

Environment triggers specify when and how a deployment will be triggered on an environment.

Environment	Trigger
Dev	Manual
QA	Manual
Production	Automated: after release creation






Even though a release is automatically created, it might not be deployed automatically to an environment. To enable automatic deployment, **triggers** must also be configured in each environment.


Release Definitions – Release names

Definition*: FabrikamDev | [Releases](#)

Environments Artifacts Configuration Triggers **General** History

 |  Save |  Release ▼

Define the naming scheme for releases of this release definition. View the list of available [pre-defined variables](#).


Release name format 

The names of releases for a release definition are, by default, sequentially numbered. The first release is named Release-1, the next release is Release-2, and so on. The naming scheme can be changed by editing the release name format mask




Release Definitions – Release names


Variable	Description
Rev:rr	An auto-incremented number with at least the specified number of digits.
Date / Date:MMddyy	The current date, with the default format MMddyy. Any combinations of M/MM/MMM/MMMM, d/dd/ddd/dddd, y/yy/yyyy/yyyy, h/hh/H/HH, m/mm, s/ss are supported.
System.TeamProject	The name of the team project to which this build belongs.
Release.ReleaseId	The ID of the release, which is unique across all releases in the project.
Release.DefinitionName	The name of the release definition to which the current release belongs.
Build.BuildNumber	The number of the build contained in the release. If a release has multiple builds, this is the number of the build that triggered the release in the case of continuous deployment, or the number of the first build in the case of a manual trigger.
Build.DefinitionName	The definition name of the build contained in the release. If a release has multiple builds, this is the definition name of the build that triggered the release in the case of continuous deployment, or the definition name of the first build in the case of manual trigger.
Artifact.ArtifactType	The type of the artifact source linked with the release. For example, this can be Team Build or Jenkins.
Build.SourceBranch	The branch for which the build in the release was queued. For Git, this is the name of the branch in the form refs/heads/master. For Team Foundation Version Control, this is the root server path for the workspace in the form \$/teamproject/branch. This variable is not set in the case of Jenkins artifact sources.
Custom variable	The value of a global configuration property defined in the release definition.


Release Definitions – Release Retention

Definition: FabrikamDev  | [Releases](#)

Environments Artifacts Configuration Triggers **General** History

 |  Save |  Release ▼

Release name format 

Release retention policy 

The release retention setting controls how long a release is retained. By default, this is 60 **days (Maximum 365)**. Releases that have not been deployed or modified during that time will automatically be deleted.

Release Definitions – Security

- **Permissions** define the **authorizations** that can be granted or denied to **users** and **groups**.
- Permissions can be granted or denied in a **hierarchical model** at the team **project level**, for a **specific release definition**, or for a **specific environment** in a release definition.
- Permissions can be **inherited** from the parent or **overridden**.

Release Definitions – Security

Control panel > DefaultCollection > Fabrikam

Overview Iterations Areas **Security** Alerts Version Control Service Hook

Create VSTS group

Filter users and groups

- Teams
 - FabrikamDev Team
- VSTS Groups
 - Build Administrators
 - Contributors
 - Endpoint Administrators
 - Endpoint Creators
 - Project Administrators
 - Project Valid Users
 - Readers
 - Release Administrators**

FabrikamDev > Release Administrators | Edit...

Permissions Members Member of

Members of this group can perform all operations on Release Management

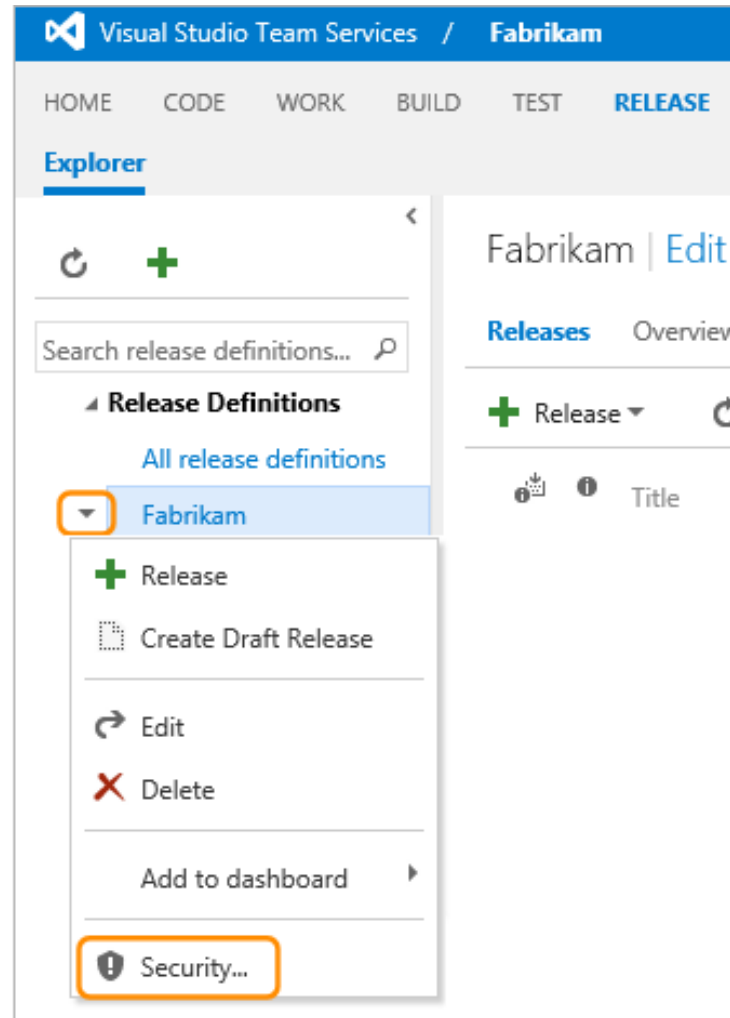
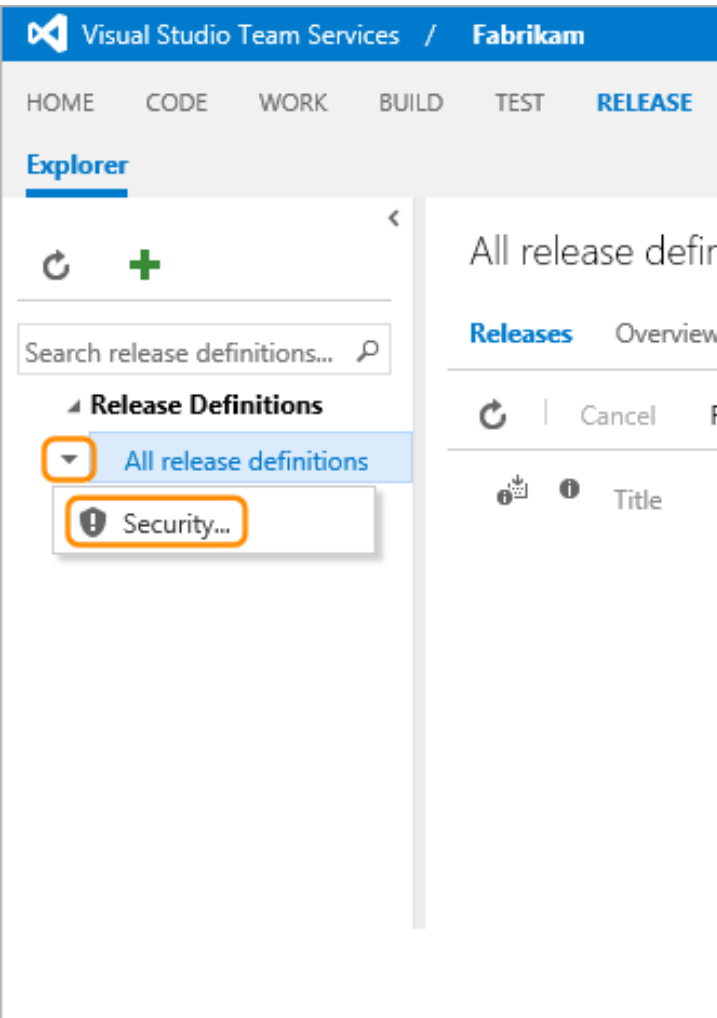
Create tag definition	Not set
Create test runs	Not set
Delete team project	Not set
Delete test runs	Not set
Edit project-level information	Not set
Manage test configurations	Not set
Manage test environments	Not set
View project-level information	Not set
View test runs	Not set

Clear explicit permissions

Save changes Undo changes

Release Management defines a Team Services group named **Release Administrators** at project level. You can add project members to this group, and set permissions for group members.

Release Definitions – Security



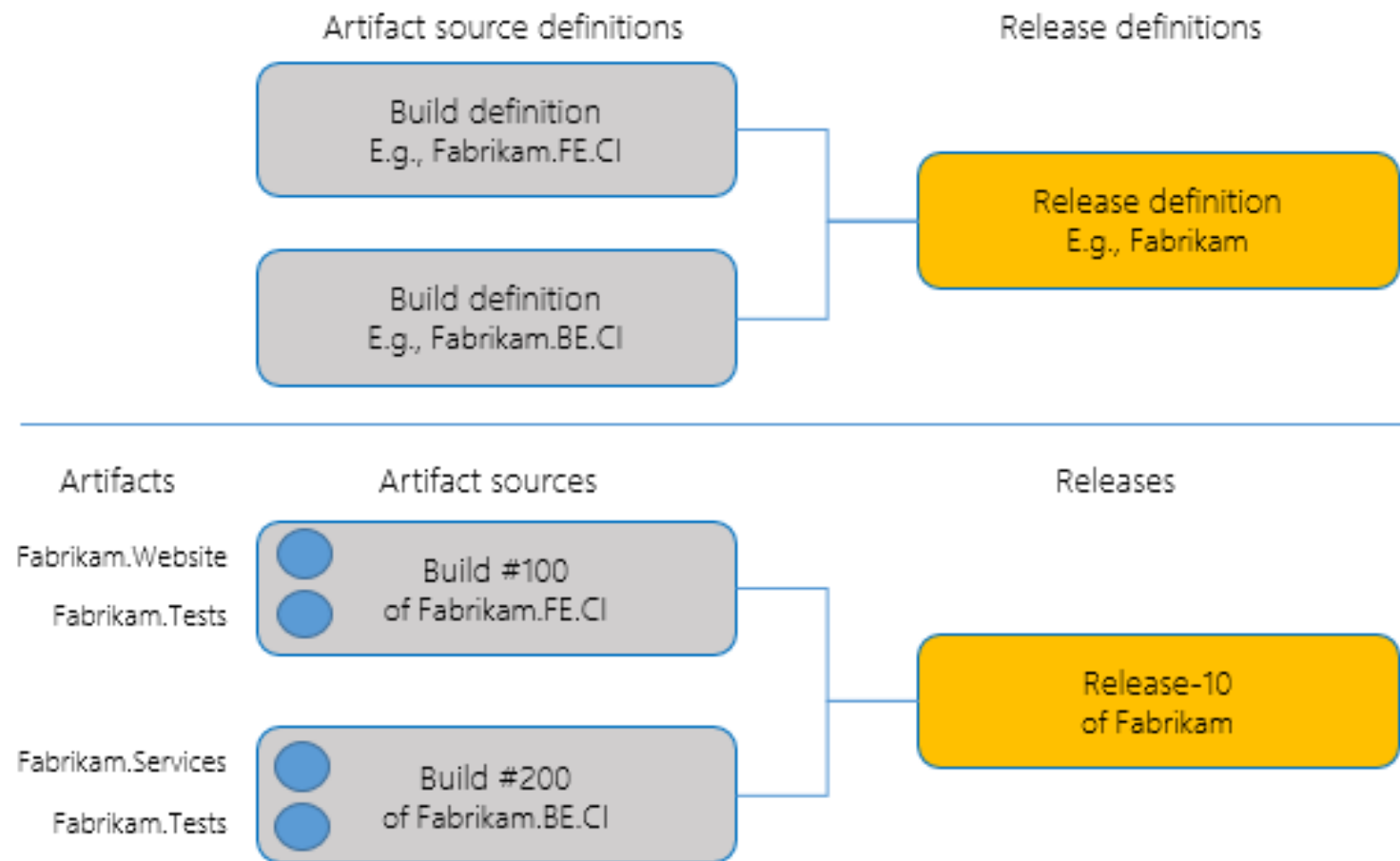
Permissions can be set for either **ALL** Release Definitions in a project, or for a **SPECIFIC** one.

Release Definitions – Security

Permission	Description	Scopes
Administer release permissions	Can change any of the other permissions listed here.	Project, Release definition, Environment
Edit release definition	Can save any changes to a release definition, including configuration variables, triggers, artifacts, and retention policy as well as configuration within an environment of the release definition. To make changes to a specific environment in a release definition, the user also needs Edit release environment permission.	Project, Release definition
Edit release environment	Can edit environment(s) in release definition(s). To save the changes to the release definition, the user also needs Edit release definition permission. This permission also controls whether a user can edit the configuration inside the environment of a specific release instance. The user also needs Manage releases permission to save the modified release.	Project, Release definition, Environment
Manage releases	Can edit the configuration in releases. To edit the configuration of a specific environment in a release instance, the user also needs Edit release environment permission.	Project, Release definition
Delete release definition	Can delete release definition(s).	Project, Release definition
Delete release environment	Can delete environment(s) in release definition(s).	Project, Release definition, Environment
Delete releases	Can delete releases for a definition.	Project, Release definition
Manage release approvers	Can add or edit approvers for environment(s) in release definition(s). This permissions also controls whether a user can edit the approvers inside the environment of a specific release instance.	Project, Release definition, Environment
Queue releases	Can create new releases.	Project, Release definition
Manage deployments	Can initiate a direct deployment of a release to an environment. This permission is only for direct deployments that are manually initiated by selecting the Deploy action in a release. If the condition on an environment is set to any type of automatic deployment, the system automatically initiates deployment without checking the permission of the user that created the release.	Project, Release definition, Environment
View release definition	Can view release definition(s).	Project, Release definition
View releases	Can view releases belonging to release definition(s)	Project, Release definition

Understanding Release Artifacts

Release Definitions – Artifacts



When **authoring** a release definition, artifact sources are **linked** release definitions.

When **creating** a release, the exact version of these artifacts must be specified.


Release Definitions – Linking Artifacts


CREATE NEW RELEASE DEFINITION

×

Artifacts

Choose a source that publishes the artifact to be deployed


Build


Jenkins

Choose Later

Project

Fabrikam

Source (Build definition)

ASPNet4.CI

☐ Continuous deployment (create release and deploy whenever a build completes)

Queue

Select an agent queue | [manage queues](#)

Hosted

< Previous

Create

Cancel

When authoring a release definition, **Artifacts** can be linked to a definition.

Release Definitions – Linking Artifacts

Definition*: FabrikamDev

Environments | Artifacts | Configuration | Triggers | General | History

🔄 | 📁 Save | + Release ▾


FabrikamDev

⚠ No build definitions are linked to this release definition. [Link to a build definition](#)

+ Add environments

Dev ...
0 👤 | 7 🔄

+ Add tasks

 **Azure Web App Deployment ***
Deploy Website to Azure ✕

Deploy Web

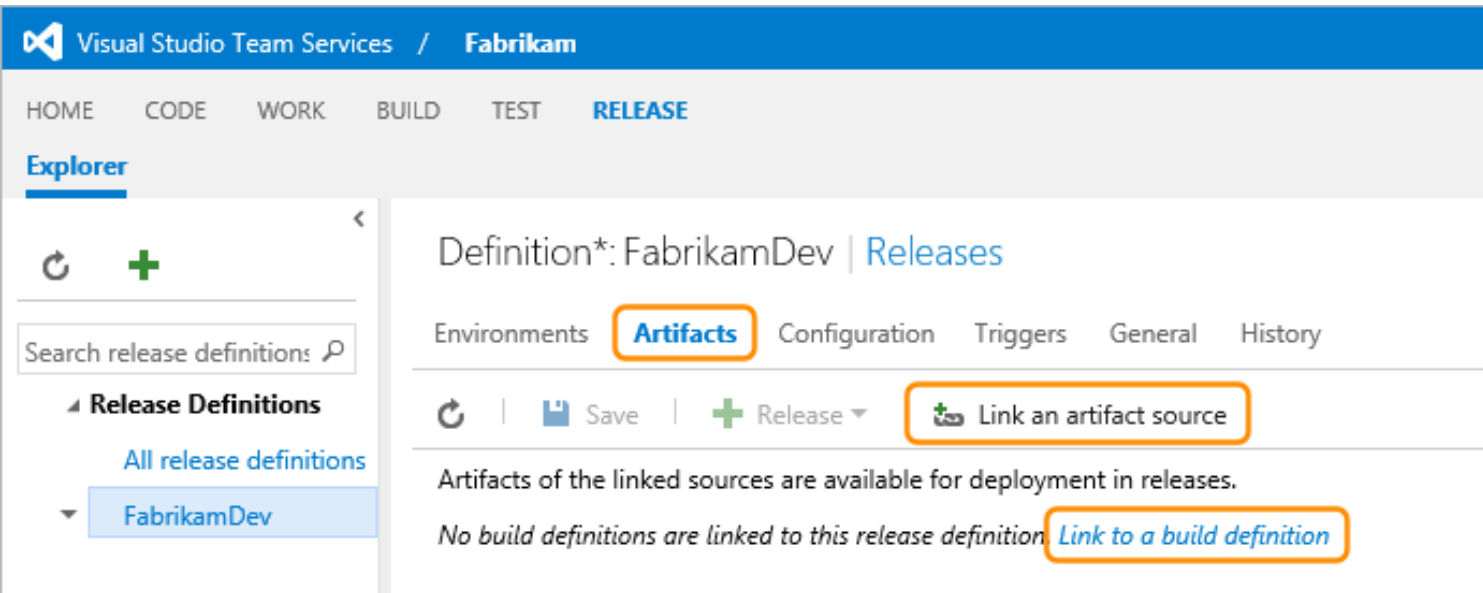
Azure Subscr

Web App Na

Web App Loc

Or ...

Release Definitions – Linking Artifacts



Visual Studio Team Services / Fabrikam

HOME CODE WORK BUILD TEST **RELEASE**

Explorer

Search release definitions 🔍

Release Definitions

All release definitions

FabrikamDev

Definition*: FabrikamDev | Releases

Environments **Artifacts** Configuration Triggers General History

🔄 | 📁 Save | + Release ▾ **🔗 Link an artifact source**

Artifacts of the linked sources are available for deployment in releases.

No build definitions are linked to this release definition. [Link to a build definition](#)

Or ...

Release Definitions – Artifacts Download

- When a release is deployed to an environment, the versioned **artifacts** from each of the sources are, by default, **downloaded to the automation agent**. Tasks running within that environment can then deploy the artifacts.
- Accessible using the Variable: **\$(Agent.ReleaseDirectory)**
- **Every time** a release definition is deployed, the previously downloaded artifacts are **deleted** and the required set of artifacts are **downloaded again**.
- To ensure this uniqueness, each artifact source linked to a release definition is automatically provided with a specific download location known as the *source alias* :
\$(Agent.ReleaseDirectory)\[source alias]


Release Definitions – Artifact Variables



- \$(RELEASE.ARTIFACTS.[source-alias].[variable-name])

variable-name	Meaning	Team Build	Jenkins	TeamCity	TFVC or Git
DEFINITIONID	The identifier of the build definition or repository.	Yes	No	No	No
DEFINITIONNAME	The name of the build definition or repository.	Yes	No	No	Yes
BUILDNUMBER	The build number or the commit identifier.	Yes	Yes	Yes	Yes
BUILDID	The build identifier.	Yes	Yes	Yes	No
REPOSITORY_NAME	The name of the repository from which the source was obtained.	Yes	No	No	No
REPOSITORY_PROVIDER	The type of repository from which the source was built; for example, TfsGit.	Yes	No	No	No
SOURCEBRANCH	The path of the branch from which the source was built.	Yes	No	No	No
SOURCEBRANCHNAME	The name of the branch from which the source was built.	Yes	No	No	No
SOURCEVERSION	The commit that was built.	Yes	No	No	No
BUILDURI	The URL for the build.	Yes	No	No	No
REQUESTEDFORID	The identifier of the account that triggered the build.	Yes	No	No	No
REQUESTEDFOR	The name of the account that requested the build.	Yes	No	No	No
TYPE	The type of artifact source, such as Team Build.	Yes	Yes	Yes	Yes

Release Definitions – Artifact Variables

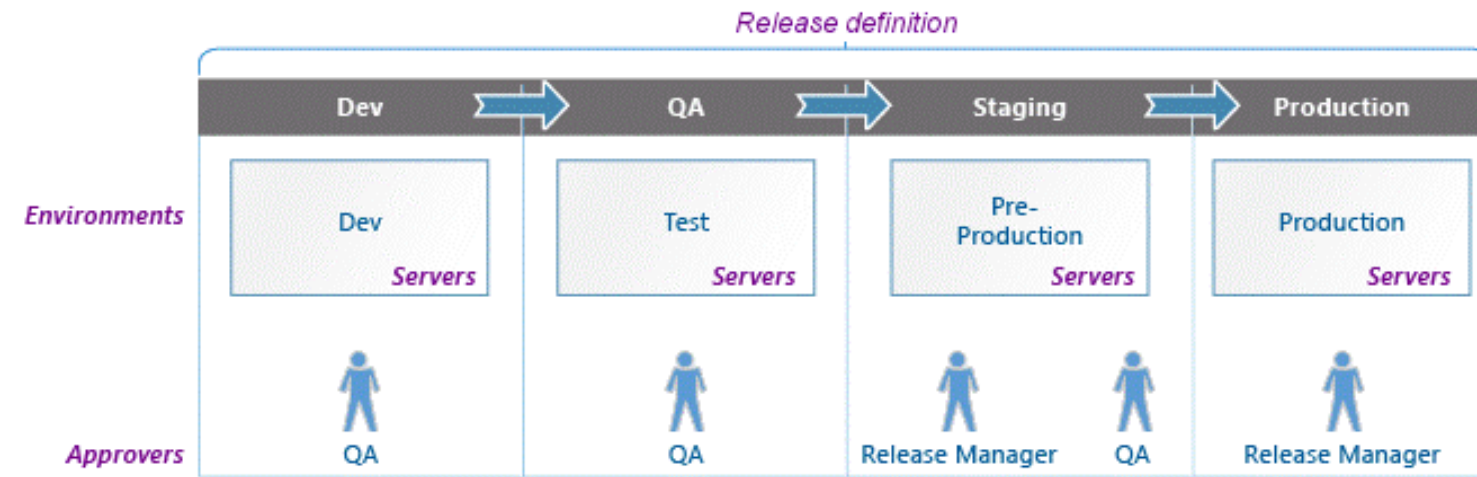
- `$(RELEASE.ARTIFACTS.[source-alias].[variable-name])`

Powershell Script 

Type	<input type="text" value="Inline Script"/>	
Arguments	<input type="text" value="-a '\$(Release.Artifacts.ASPNET4.CI.DEFINITIONNAME)'"/>	
Inline Script	<pre>param (\$a) Write-Host "\$a"</pre>	

Understanding Release Environments

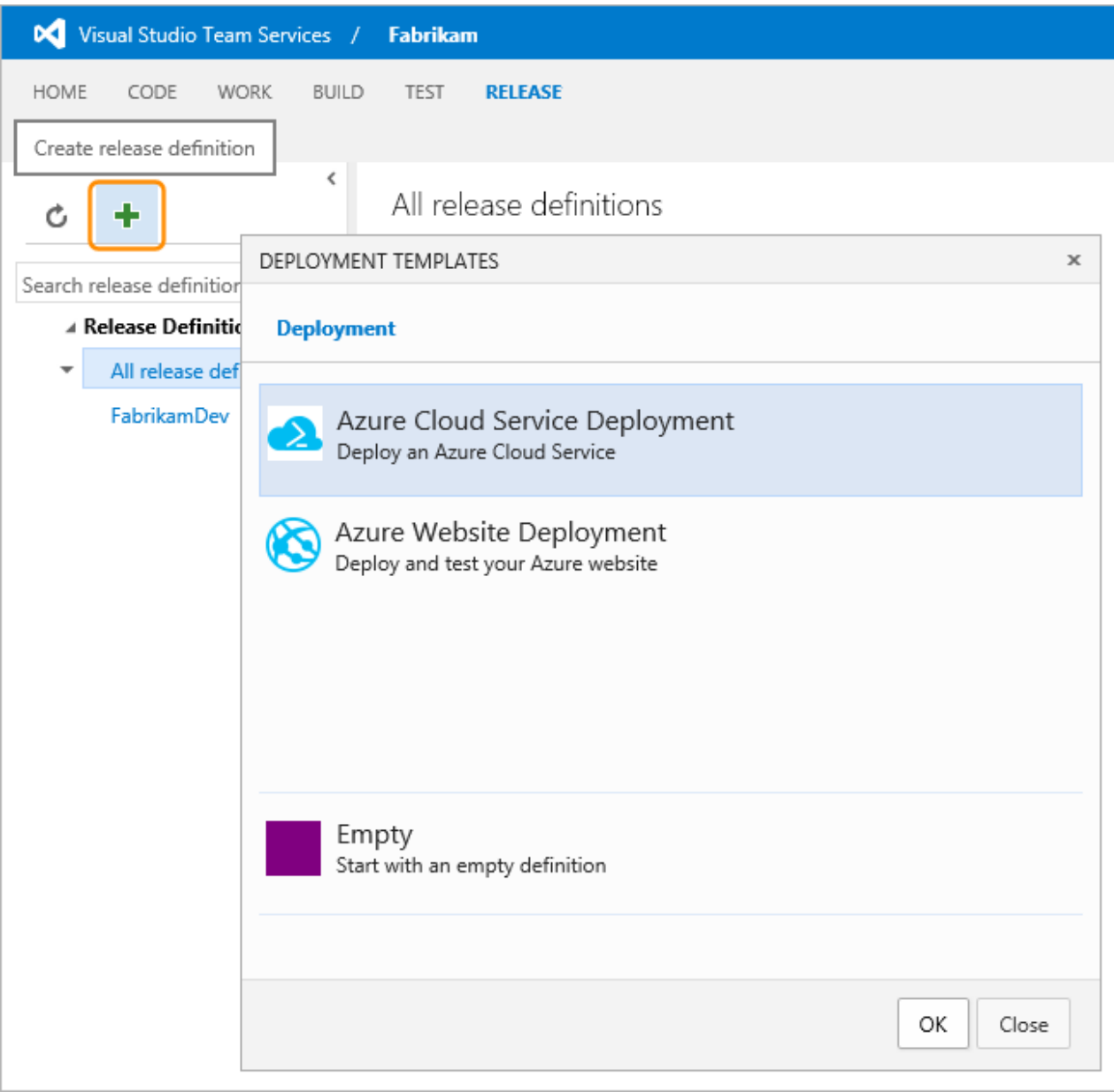
Release Definitions – Environments



An environment is a **logical** entity that represents **where to deploy** a release.

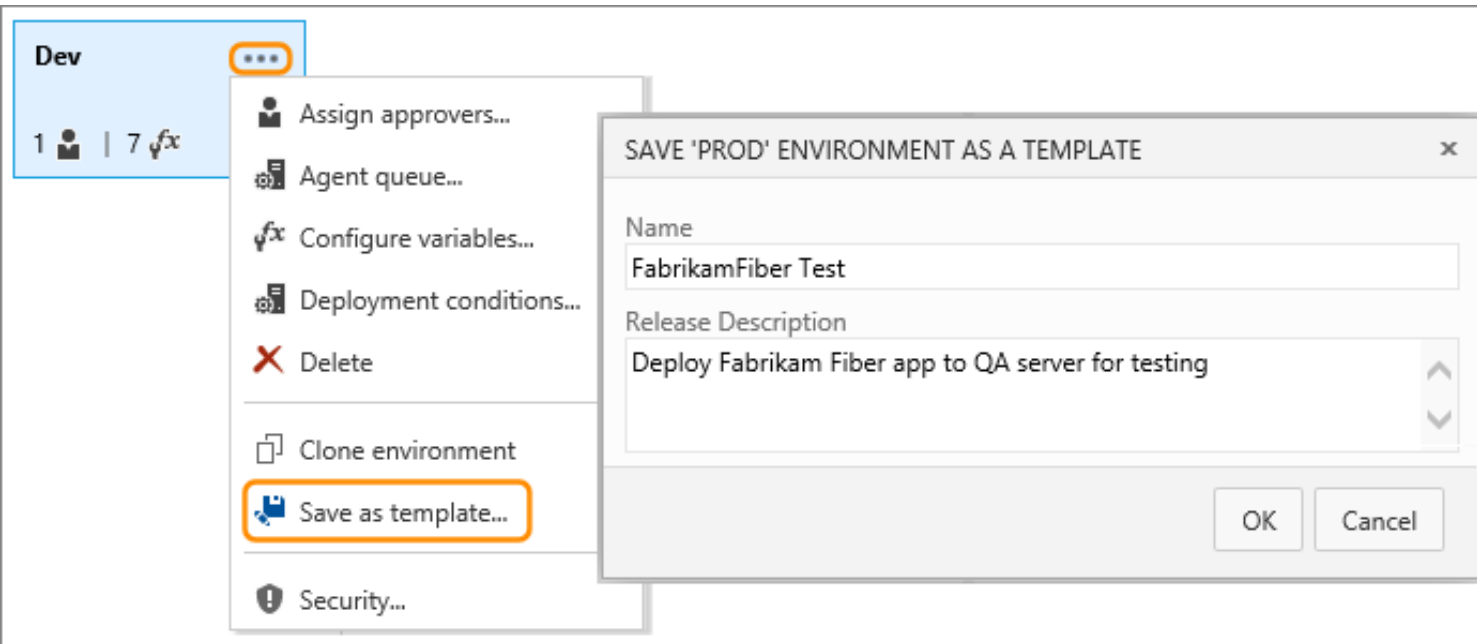
Physically, the deployment in an environment may happen to a collection of servers, a cloud, multiple clouds, or an app store

Release Definitions – Environment Templates



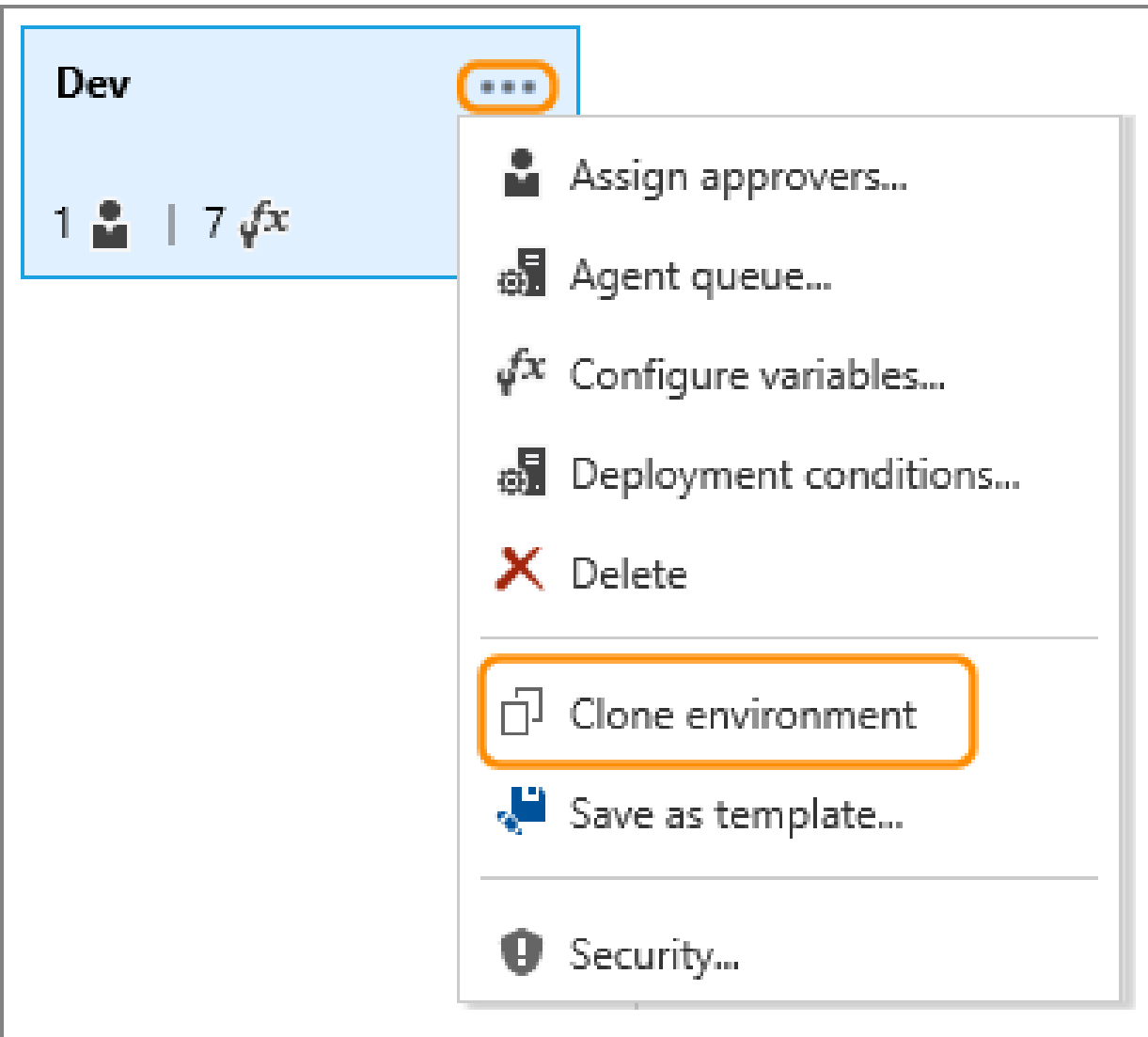
To initiate a new release definition, or to add an environment to an existing release definition, a deployment templates which are pre-populated with tasks and settings, can be used for each environment.

Release Definitions – Environment Templates



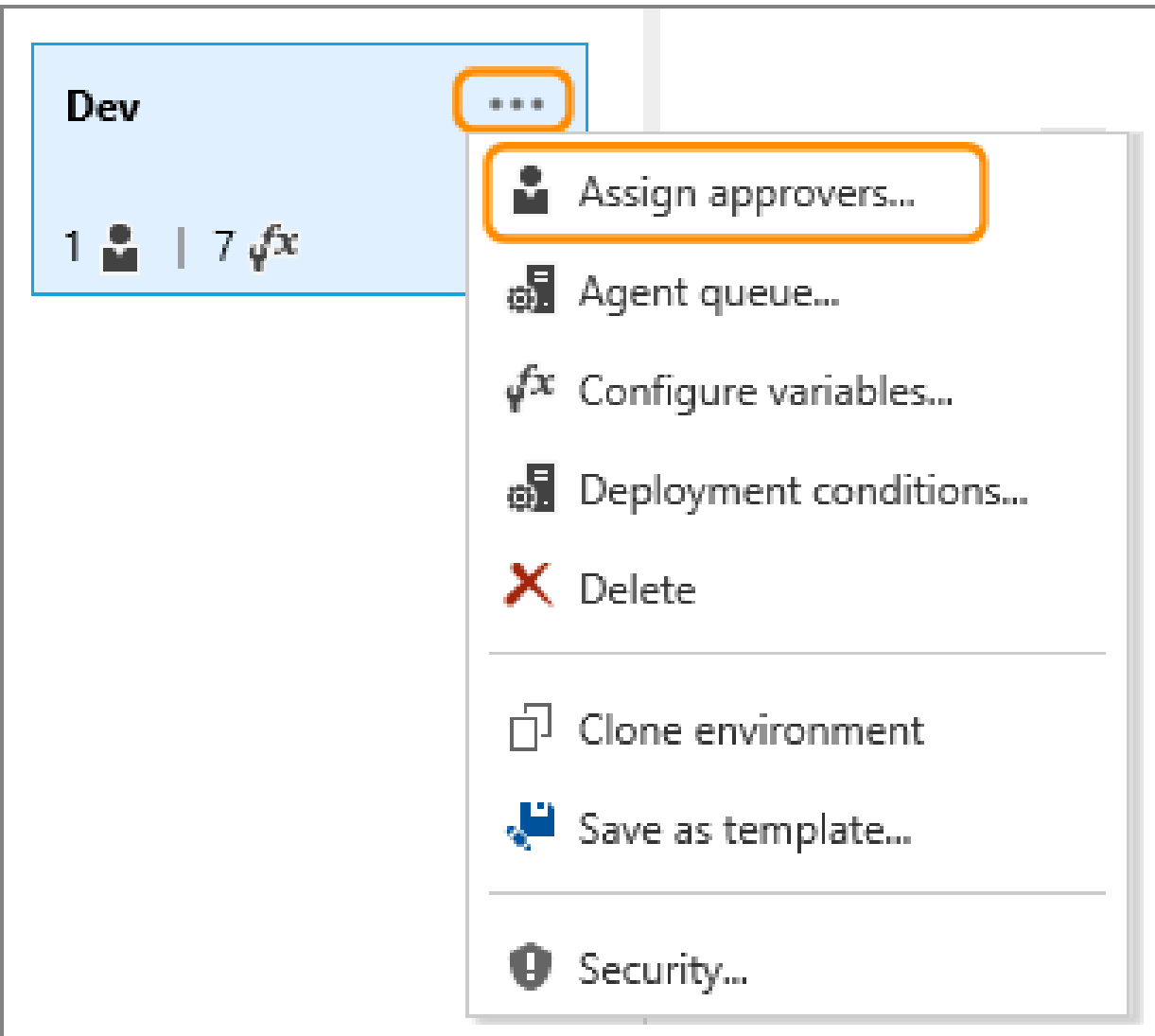
Custom environment templates can be created and saved as **templates**.

Release Definitions – Environment Templates



After an environment is added to a release definition and configured it by adding tasks and setting the properties for each one, it can be **cloned** it to create another environment within the same definition.

Release Definitions – Environment Approvals



Approvers can be defined for each environment in a release definition. At each point where approval is required until the specified approver **grants approval or rejects** the release (or **re-assigns** the approval to another user)

Release Definitions – Environment Approvals

CONFIGURE - 'DEV' ENVIRONMENT

Approvals

Queue

Variables

General

Deployment conditions

Approvers

Select the users who can approve or reject the deployments to this environment. ⓘ

Pre-deployment approver

☒ Automatic

☐ Specific Users

Post-deployment approver

☒ Automatic

☐ Specific Users

Options

Apply these rules for the deployment approvers

☐ Send an email notification to the approver whom the approval is pending on

☐ The user who creates the release cannot approve

OK

Cancel

Approvals can be automated as well.

Release Definitions – Environment Approvals

CONFIGURE - 'DEV' ENVIRONMENT


Approvals Queue Variables General Deployment conditions

Approvers

Select the users who can approve or reject the deployments to this environment. ⓘ

Pre-deployment approver ☐ Automatic ☒ Specific Users

Rai

 **Raisa Pokrovskaya** ×
raiskovskaya@msn.com

Post-deployment approver ☒ Showing 1 results ☐ Specific Users

Options

Apply these rules for the deployment approvers

☐ Send an email notification to the approver whom the approval is pending on

☐ The user who creates the release cannot approve

OK Cancel

There can be **one or many** approvers for both pre and post deployments.

Release Definitions – Environment Approvals

CONFIGURE - 'DEV' ENVIRONMENT ×

[Approvals](#) Queue Variables General Deployment conditions

Approvers

Select the users who can approve or reject the deployments to this environment. ⓘ

Pre-deployment approver ☐ Automatic ☒ Specific Users

Raisa Pokrovskaya ✕ Mateo Escobedo ✕ Search users and groups

☒ All of the specified users in any order
☐ Any one of the specified users
☐ All of the specified users in sequential order

[Fewer Options](#)

Approvals can also follow a specific **sequential** order.

Release Definitions – Environment Approvals

CONFIGURE - 'DEV' ENVIRONMENT

Approvals Queue Variables General Deployment conditions

Approvers

Select the users who can approve or reject the deployments to this environment. ⓘ

Pre-deployment approver ☐ Automatic ☒ Specific Users

[Fabrikam]\Fabrikam Team X Search users and groups

[More Options](#)

Post-deployment approver ☒ Automatic ☐ Specific Users

Options

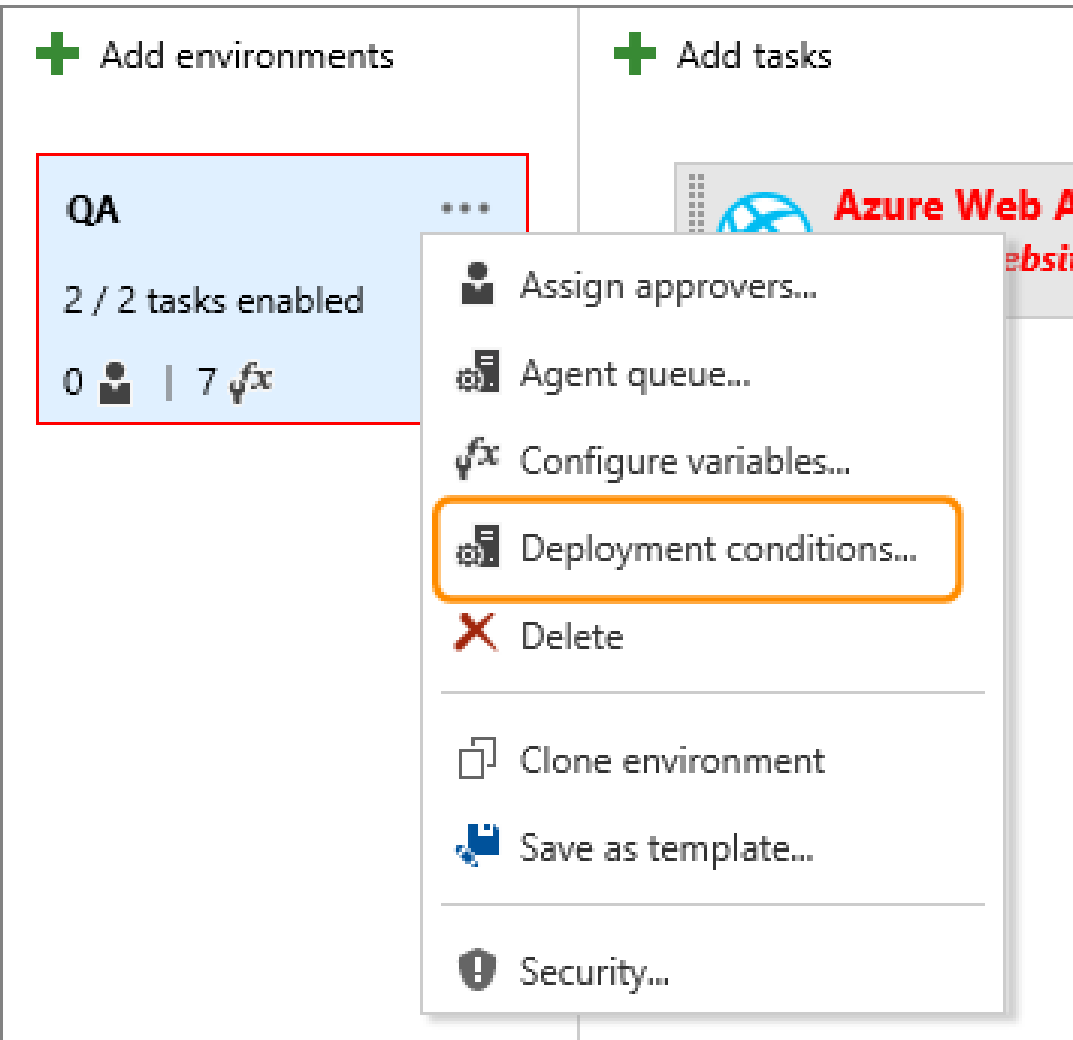
Apply these rules for the deployment approvers

- ☒ Send an email notification to the approver whom the approval is pending on
- ☒ The user who creates the release cannot approve

OK Cancel

Approvals can be requested from a team.

Release Definitions – Environment Deployment Triggers



Deployment **triggers** must be configured for each environment in the definition in order for the new release to be **deployed automatically** to one or more of the environments in a release definition.

Release Definitions – Environment Deployment Triggers

CONFIGURE - 'DEV' ENVIRONMENT

Approvals Queue Variables General Deployment conditions

Trigger

Define when the deployment on this environment should be started automatically. If you want to deploy manually to this environment, use the "Deploy" action.

☐ No automated deployment, manual only

☒ After release creation

☐ After successful deployment

Options

Define behavior when multiple releases are waiting to be deployed on this environment. ⓘ

☒ Allow multiple releases to be deployed at the same time

☐ Allow only one active deployment at a time

OK Cancel

- After release creation.

Release Definitions – Environment Deployment Triggers

CONFIGURE - 'PRODUCTION' ENVIRONMENT

Approvals Queue Variables General **Deployment conditions**

Trigger

Define the trigger that will start deployment to this environment.

☐ No automated deployment

☐ After release creation

☒ After successful deployment on another environment

Trigger a new deployment on this environment after successful deployment on the selected environment.

Triggering environment

☐ Dev

☒ QA

☐ Pre-prod

Options

Define behavior when multiple releases are waiting to be deployed on this environment. ⓘ

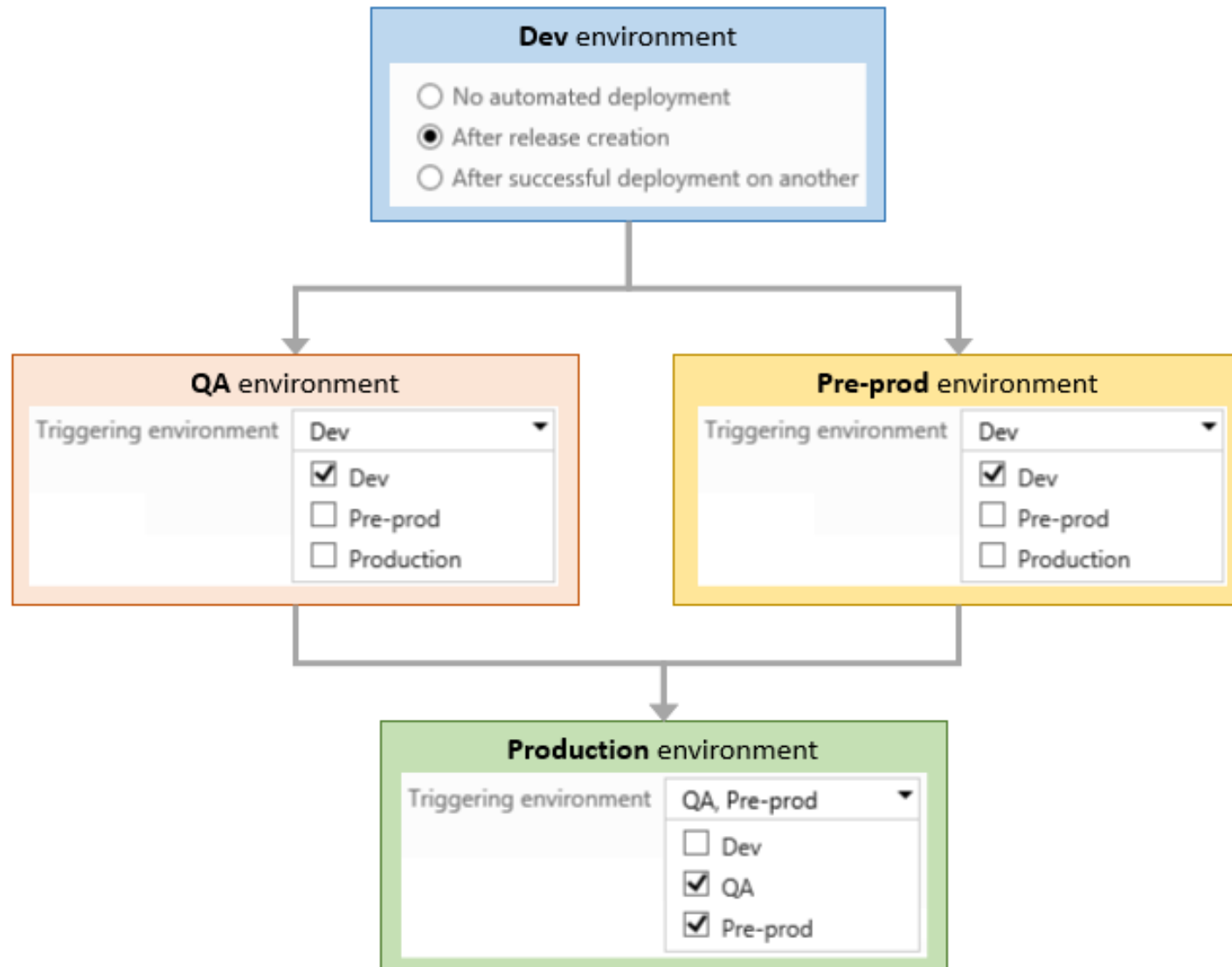
☒ Allow multiple releases to be deployed at the same time

☐ Allow only one active deployment at a time

OK Cancel

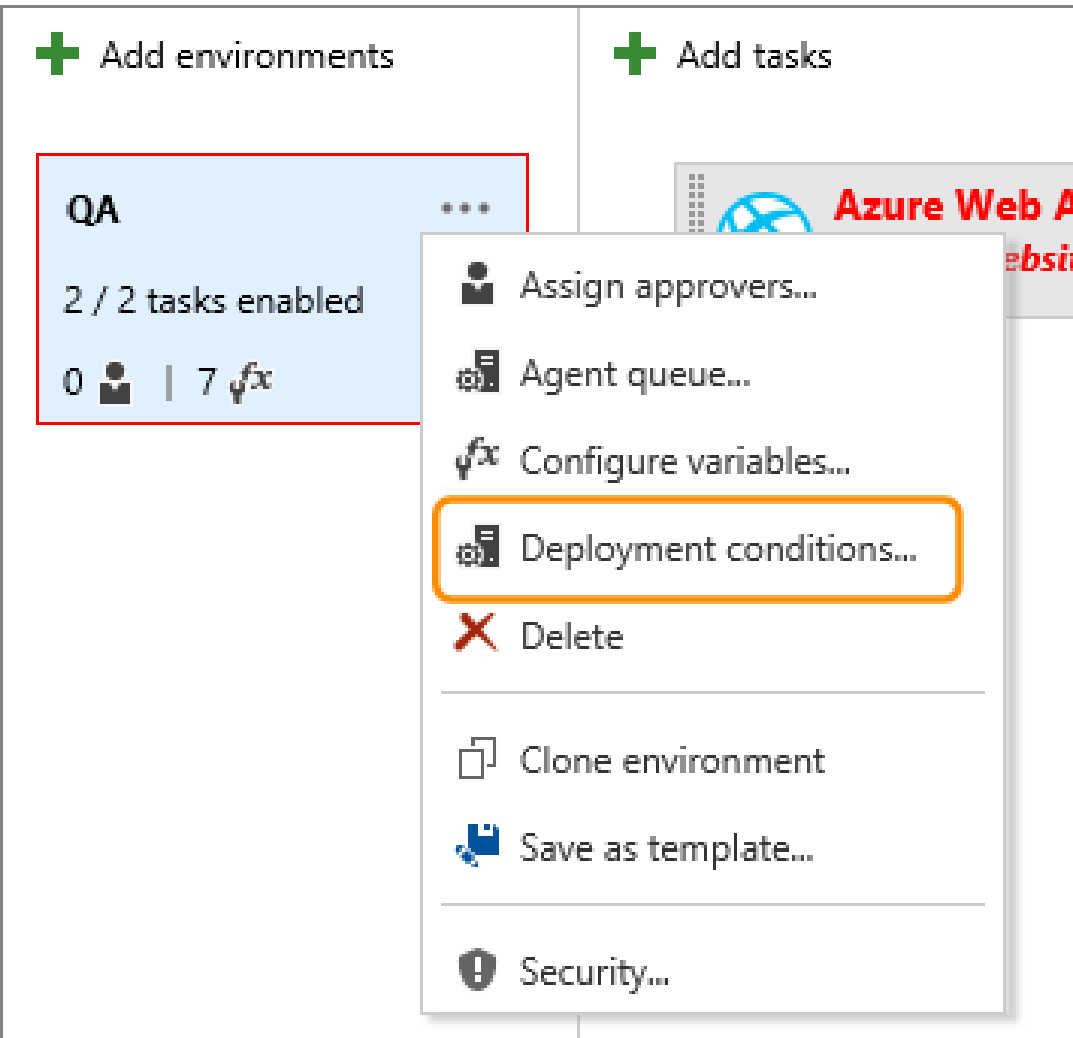
- After successful deployment.

Release Definitions – Parallel forked and joined deployments



The Triggering environment list allows selecting more than one environment. This allows configuring parallel (forked and joined) deployment pipelines where the deployment to an environment occurs only when deployment to all the selected environments succeeds

Release Definitions – Queuing policies



It is possible to control how multiple releases are queued into an environment.

Release Definitions – Queuing policies

CONFIGURE - 'DEV' ENVIRONMENT

Approvals Queue Variables General Deployment conditions

Trigger

Define when the deployment on this environment should be started automatically. If you want to deploy manually to this environment, use the "Deploy" action.

☐ No automated deployment, manual only

☒ After release creation

☐ After successful deployment

Options

Define behavior when multiple releases are waiting to be deployed on this environment. ⓘ

☐ Allow multiple releases to be deployed at the same time

☒ Allow only one active deployment at a time

For subsequent deployment requests:

☐ Deploy all of them one after another

☒ Deploy only the latest release

OK Cancel

Options are:

- Allow multiple releases to be deployed at the same time.
- Allow only one active deployment at a time:
 - Latest
 - One after the other.

Release Definitions –Configuration Properties

Can be defined as **custom variables** that are global to a release definition, and can be **used** to specify values for **any of the tasks** in any of the environments of a release definition.

Release Definitions –Configuration Properties




- Global configuration properties should be used when the same value is needed across all the environments.
- Environment-level configuration properties should be used for values that vary from environment to environment.
- Properties can be encrypted. The values of such properties are stored securely on the server and cannot be viewed by users once they are saved.

Release Definitions –Configuration Properties

Definition*: Purchasing | Releases

Environments | Artifacts | **Configuration** | Triggers | General | History




Release variables


 |  Save |  Release ▼

Environment variables

Release variables

Define custom variables to use in this release definition. View list of [pre-defined variables](#)

Name	Value
AzureContainerUri 
AzureStorageAccount	fabrikaminfra 
AzureStorageContainer	purchasing 

 **Add variable**

Release Definitions –Configuration Properties

Definition: FabrikamDev | [Releases](#)

Environments

Artifacts

Configuration

Triggers

General

↺

 |

💾 Save

 |

⊕ Release ▼

Release variables

Environment variables

Release variables

View and modify custom variables defined on the environments in this release definition.

	Dev	Test	Production
AdministratorLogin	<input type="text"/>	<input type="text"/>	<input type="text"/>
AdministratorLoginPassword	<input type="password" value="*****"/>	<input type="password" value="*****"/>	<input type="password" value="*****"/>
ConnectionStringName	<input type="text"/>	<input type="text"/>	<input type="text"/>
DatabaseName	<input type="text"/>	<input type="text"/>	<input type="text"/>
ReleaseConfiguration	<input type="text" value="Release"/>	<input type="text" value="Release"/>	<input type="text" value="Release"/>
ReleasePlatform	<input type="text" value="Any CPU"/>	<input type="text" value="Any CPU"/>	<input type="text" value="Any CPU"/>

Understanding Release Management Tasks


Release Definitions – Tasks


- Release definitions execute tasks.
- TFS and VSTS ship with many out of the box tasks.
- These tasks can be extended:
 - Scripts. (Any)
 - From the Market place
 - By Creating Extensions.
 - Endpoints.


Release Definitions – Tasks


ADD TASKS


All
Build
Utility
Test
Package
Deploy


 Deploy Azure SQL DB using DACPAC


 Azure Web App Deployment
Publish a Visual Studio Web project to a Microsoft Azure Web App using Web Deploy


 Azure Web Deploy
Invokes web deploy to Azure website, substituting any Environment variables into the SetParameters.xml file

 PowerShell on Target Machines
Execute PowerShell scripts on remote machine(s)

 Rollback PowerShell
Run a powershell script to rollback deployments. Task execution context is available in the powershell context for implementing conditional rollback

 Windows Machine File Copy
Copy files to remote machine(s)

 WinRM - IIS Web App Deployment
Connect via WinRM, to deploy Web project locally on IIS, using Web Deploy

 WinRM - IIS Web App Management
Connect via WinRM, to create or update the Website and App Pool locally on IIS

Add

Add

Add

Add

Add

Add


Add


Add


Close


ADD TASKS


All
Build
Utility
Test
Package
Deploy


 Publish build artifacts to the server or a file share


 Replace Tokens
Replaces tokens in a file using RegEx. Values come from any variable defined in the current Environment.


 Replace Tokens
Replace tokens in files


 Shell Script
Run a shell script using bash

 Shell++
Execute bash script

 Tokenize with XPath/Regular expressions
Replaces __<variable>__ and/or XPath for XML documents with User Defined variables or configuration json document

 UnZip A package
UnZip a package

 Xamarin License
Activate or deactivate Xamarin licenses

 Zip A Folder
Zip Files and Folder

Add

Add

Add

Add

Add

Add

Add

Add

Release Definitions – Endpoints

The screenshot shows the 'Endpoints' page in the Azure DevOps interface. The breadcrumb navigation at the top reads 'Control panel > DefaultCollection > Fabrikam'. Below this is a horizontal menu with tabs: 'Overview', 'Iterations', 'Areas', 'Security', 'Alerts', 'Version Control', 'Service Hooks', and 'Services' (which is highlighted with an orange border). The main content area is titled 'Endpoints' and 'XAML Build Services'. On the left, there is a '+ New Service Endpoint' button with a dropdown menu. The dropdown menu is open, showing a list of service types: 'Generic', 'External Git', 'GitHub', 'Subversion', 'Azure Classic', 'Azure Resource Manager', 'Chef', and 'Jenkins'. To the right of the dropdown is a search icon. Further right, under the heading 'Services', it says 'No Services found.'

Endpoints allow release manage to connect to any source in order to get artifacts or to deploy them.