**« Microsoft Azure DevOps (Session10)»**

**معاونت** توسعه و برند

واحد نرم‌افزار



فهـرســت

[Pipeline Azure 2](#_Toc73170654)

[CI (Continuous Integration) 3](#_Toc73170655)

[CD (Continuous Delivery (Deployment)) 3](#_Toc73170656)

[کانسپت های کلیدی 4](#_Toc73170657)

[Trigger 4](#_Toc73170658)

[Stage 4](#_Toc73170659)

[Approval 5](#_Toc73170660)

[job 6](#_Toc73170661)

[Agent 6](#_Toc73170662)

[Pipeline 6](#_Toc73170663)

[Environment 6](#_Toc73170664)

[Deployment group 7](#_Toc73170665)

[Step 7](#_Toc73170666)

[Script 7](#_Toc73170667)

[Task 8](#_Toc73170668)

[Run 8](#_Toc73170669)

[Artifact 8](#_Toc73170670)

[استفاده از Classic interface 8](#_Toc73170671)

[استفاده از YAML syntax 9](#_Toc73170672)

[دانلود و نصب Agent جهت اجرای Pipeline: 10](#_Toc73170673)

[آشنایی با بخش Task groups 11](#_Toc73170674)

[آشنایی با بخش Library 11](#_Toc73170675)



# Pipeline Azure

جهت پیاده سازی و یکپارچه سازی CI/CD (ادغام و تحویل مداوم) و با هدف اجرای خودکار فرآیندهای Test، Build و Delivery سورس کدها – با هر زبان برنامه نویسی – به صورت منظم و پیوسته، مورد استفاده قرار می‌گیرد.

# CI (Continuous Integration)

این فرآیند کمک می‌کند تا تغییرات اعمال شده بر روی سورس کد دائما با ریپازیتوری اصلی ادغام گردد. این مهم پس از کسب اطمینان از عدم ناسازگاری مابین کدها یا کامپوننت‌های قدیمی و جدید که توسط تعدادی تست بررسی می‌گردند، نهایی می‌شود و با نام Build pipeline یاد می‌شوند.

**نکته:** همواره فرآیند Build مرتبط به یک ریپازیتوری و یک انشعاب واحد است.

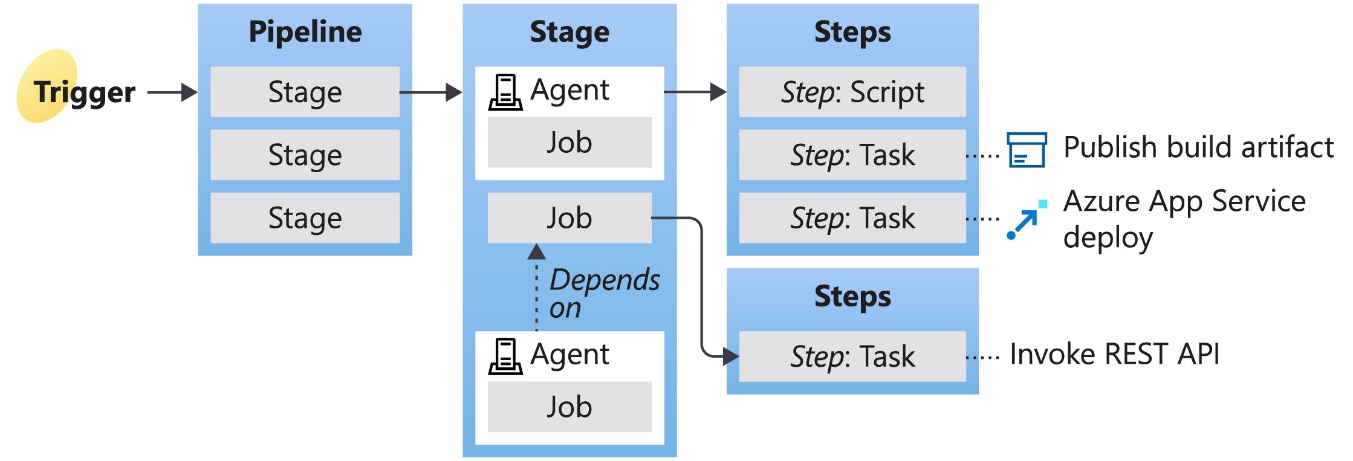
# CD (Continuous Delivery (Deployment))

این امکان را به توسعه‌دهندگان می‌دهد تا به صورت روزانه،‌ هفتگی و یا ماهانه نسخهٔ جدید نرم‌افزار خود را بصورت دستی یا خودکار بر روی سرور هدف و در دسترس کاربران قرار دهند و با نام Release pipeline یاد می‌شوند.



پس می‌توان گفت Pipelineها به نوعی Automation مدیریت فعالیت‌های توسعه و انتشار پیوسته نرم افزار می‌باشند.

# کانسپت های کلیدی



# Trigger

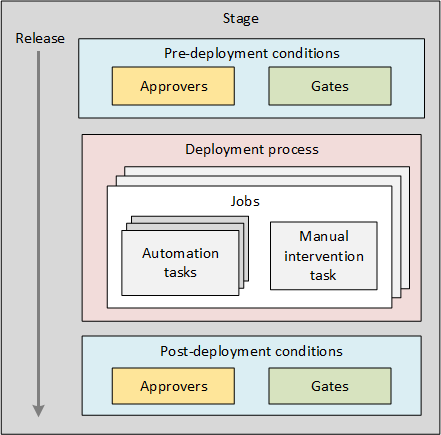
از Trigger جهت تنظیم اجرای خودکار Pipeline استفاده می‌شود.

# Sta**ge**

بعنوان یک مرز منطقی و جداکننده بخش های مختلف فرآیند Release pipeline بوده (مانند Build، QA و Production) و شامل یک یا چند Job می‌باشد. به عبارتی دیگر Stage ارائه دهنده و نماینده محیطی است که قرار است Publish روی آن اتفاق بیفتد.

**نکته:** بصورت دیفالت استیج ها بصورت پشت سر هم اجرا می‌شوند و هر استیج به یک Environment متصل می‌گردد.

**نکته:** برای ورود و خروج از هر Stage میتوان با استفاده از مفاهیم Approver و Gates، محدودیت هایی اعمال کرد. برای مدیریت Approverها بهتر است بجای استفاده از شخص یا اشخاص جهت اعتبار سنجی، از گروه‌های تعریف شده از قبل استفاده گردد. جهت اطلاعات بیشتر [کلیک](https://docs.microsoft.com/en-us/azure/devops/pipelines/release/deploy-using-approvals?view=azure-devops) فرمائید.



Approval

مجموعه ای از اعتبار سنجی های مورد نیاز قبل و بعد از فرآیند Deploy سورس کد، بر روی محیط هدف می‌باشد.



## job

هر Stage شمال یک یا چند Job و هر Job شامل یک یا چند Step یا مرحله می‌باشد. هر Job و تمامی Stepهای متعلق به آن بر روی یک Agent اجرا می‌شوند.



## Agent

## 

زمانی که Build یا Deployment ما اجرا می‌شود سیستم یک یا چند Job را با کمک Agentها استارت می‌زند. در واقع Agentها نرم افزارهایی هستند که سیستم بوسیله آنها Jobهای مورد نظر خود را اجرا می‌کند. جهت آشنایی بیشتر [کلیک](https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/agents?view=azure-devops&tabs=browser) کنید.

**نکته:** تعامل Agentها با Azure DevOps بصورت رمزنگاری شده می‌باشد.



## Pipeline

از یک یا چند Stage تشکیل شده و روند ادغام و استقرار مداوم پروژه را تعریف می‌کند. به بیان دیگر گردش کاری است که نحوه اجرای مراحل Test، Build و Deploy را تعیین کرده و بر روی یک یا چند Environment استقرار می‌یابد.

## Environment

هر Environment مجموعه ای از منابع است که بر روی آن فرآیند Deploy بصورت یکپارچه انجام می‌پذیرد و میتواند شامل یک یا چند Virtual Machine یا Kubernetes باشد.

**نکته:** برای اضافه کردن VM به Environmentها مانند Deployment Groupها عمل میکنیم. یعنی بر روی هر VM یک یا چند agent نصب میکنیم تا قابل استفاده باشد.



## Deployment group

مجموعه ای از محیط های فیزیکی یا ماشین های هدف جهت Deploy محصول می‌باشد که بر روی هر یک از آن ها Agentهایی نصب شده است. به نوعی یک Deployment group دسته بندی دیگری از Agentها (مانند Agent pool) می‌باشد.

**نکته:** با وجود Pipelineهای نسل جدید و Environmentها این مفهوم کاربرد خود را به نوعی از دست داده است چرا که Deployment groupها فقط در Classic release استفاده می‌شوند.



## Step

کوچکترین قسمت و پایین ترین لایه یک Pipeline بوده و میتواند از دو نوع Task و Script باشد.



Script

اسکریپت مجموعه ای از کدهای تعریف شده بوده که با استفاده از Command Line، PowerShell یا Bash قادر به اجرای فعالیت های خاص می باشد.



Task

هر تسک اسکریپتی پکیج شده می باشد که با استفاده از مجموعه ای از ورودی ها شکل گرفته و عملیاتی میشود.



## Run

به هربار اجرای Pipeline گویند که لاگ های Stepها و نتایج Testها را جمع آوری کرده و به محض فراخوانی، ابتدا Pipeline را پردازش کرده و متناسب با آن یک یا چند Agent را مامور به اجرا می کند.



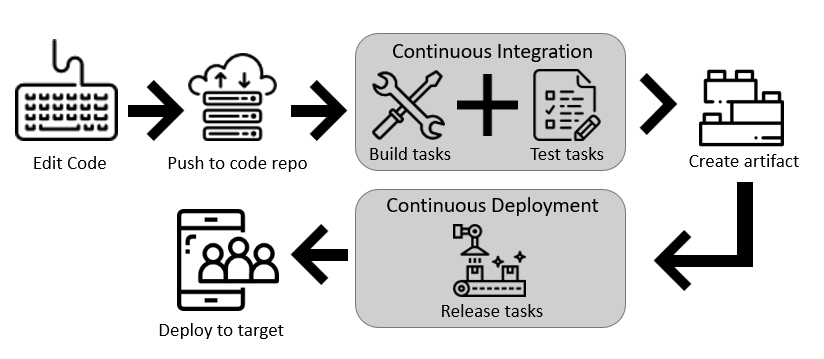
## Artifact

خروجی فرآیند CI و به عبارتی نتیجه پروژه تحت مجموعه ای از فایل ها یا پکیج های منتشر شده است که برای اقدامات بعدی مانند توزیع و استقرار (مربوط به CD) مورد استفاده قرار میگیرند.

**برای پیاده سازی فرآیندهای CI/CD با استفاده از آژور دو راه وجود دارد:**

## **استفاده از Classic interface**

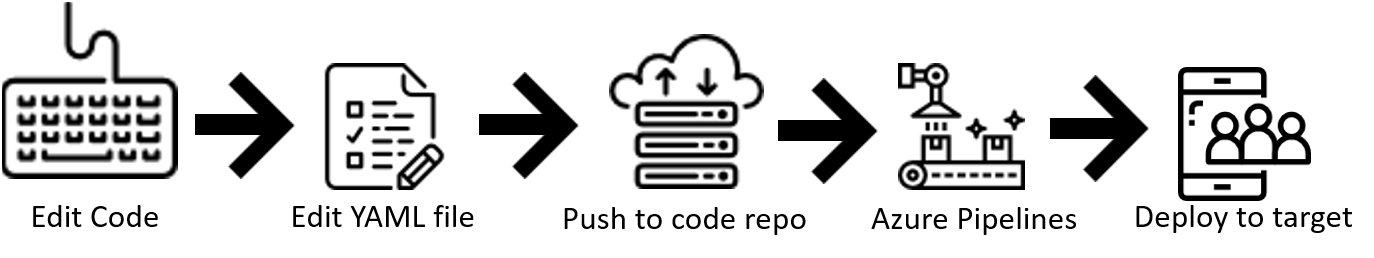
به اینصورت که با استفاده از ادیتور کلاسیک و پس از انتخاب ریپازیتوری و تمپلیت مورد نظر (استفاده از تمپلیت های نمونه یا تمپلیت خام)، یک Build pipeline برای ساخت و تست کدها ایجاد کرده و در انتها بر روی یک Artifact پابلیش میکنیم. از طرف دیگر یک Release pipeline با هدف استقرار آن Artifact بر روی سرورهای هدف ایجاد می نمائیم.



# استفاده از YAML syntax

به اینصورت که با استفاده از دستورات و تعاریف موجود در YAML (دستوراتی که مایکروسافت پشتیبانی میکند) فرآیندهای Build، Test و حتی Release خود را پیاده سازی میکنیم.

این دستورات در سندی تحت نام azure-pipelines.yml در ریپازیتوری پروژه نگهداری میشوند.



ذکر این نکته ضروری است که در مدل Classic interface پیاده سازی Build و Release بصورت جداگانه انجام می شود. این در حالیست که در مدل YAML-base میتوان علیرغم پیاده سازی جداگانه این دو فاز، Build و Release را در کنار هم و در یک فایل YAML مدیریت نمود.

* جهت درک بهتر دستورات YAML و آشنایی اولیه با این دستورات به لینک های زیر مراجعه فرمائید:
  + [YAML schema reference](https://docs.microsoft.com/en-us/azure/devops/pipelines/yaml-schema?view=azure-devops&tabs=schema%2Cparameter-schema)
  + [Understand the azure-pipelines.yml file](https://docs.microsoft.com/en-us/azure/devops/pipelines/customize-pipeline?view=azure-devops)



# دانلود و نصب Agent جهت اجرای Pipeline:

جهت دانلود Agent از مسیر زیر اقدام میکنیم:

Agent > Project Setting > Agent Pools > *Select Agent Pool* > Agents > New Agent > Download

**نکته:** بطور کلی استراتژی مدیریت Agentها و [Agent Pool](https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/pools-queues?toc=%2Fazure%2Fdevops%2Forganizations%2Ftoc.json&bc=%2Fazure%2Fdevops%2Forganizations%2Fbreadcrumb%2Ftoc.json&view=azure-devops&tabs=yaml%2Cbrowser) ها بسیار اهمیت دارد و ایجاد یک Pool به ازای هر پروژه اصلا اقدام درست و حرفه ای نیست. بنابراین با توجه به اینکه در هنگام تعریف Jobها اطلاعات مربوط به Agent pool از ما دریافت میشود، بهتر است در هر Pool چندین Agent داشته باشیم تا علیرغم جلوگیری از ایجاد Poolهای متعدد و در نتیجه مدیریت بهتر Agentها، زمان اجرای Jobها نیز کوتاه تر کرد.

**نکته:** Poolها را در دو سطح تیم پروژه و تیم کالکشن میتوان ایجاد و مدیریت کرد.  
Poolهای سطح کالکشن بصورت جنرال بوده و برای همه تیم پروژه ها به دو صورت قابل استفاده اند. صورت اول اینکه Poolها به محض ایجاد در لیست تیم پروژه ها قابل مشاهده و استفاده اند و در صورت دیگر پس از تعریف، Pool ایجاد شده در لیست تیم پروژه ها نمایش داده نمیشود و باید بصورت دستی و توسط هر تیم پروژه بر حسب صلاحدید تیمی اضافه گردد.

**نکته:** Agentها هنگام اجرا دارای حساسیت بالایی می باشند؛ به همین دلیل حین اجرای build نباید پوشه های مرتبط با آن باز و یا در حال استفاده و اعمال تغییرات (چه از طریق اکسپلور و چه ترمینال) باشند، در غیر اینصورت با خطا مواجه می شویم.



# آشنایی با بخش Task groups

Task groupها این امکان را برای ما فراهم می آورند که مجموعه ای از taskهایی که در Build pipelineها و Release pipelineها مورد استفاده قرار داده ایم را گروه بندی کرده و در سایر Pipelineهایمان مورد استفاده مجدد قرار دهیم.

**نکته:** زمانی که ما یک Task group ایجاد کرده ایم و بنابردلایلی به اعمال تغییرات در بخشی از آن گروه میپردازیم، این تغییر بطور خودکار در تمامی Pipeline های حاوی آن Task group اعمال میگردد.

**نکته:** برای ایحاد گروهی از تسک ها لازم است تمامی پارامترها بصورت Variable تعریف شده باشند تا در تمامی موقعیت ها و Pipelineها قابل استفاده باشند.

جهت کسب اطلاعات بیشتر در زمینه task groupها و آشنایی با نحوه ایجاد و مدیریت آنها [کلیک](https://docs.microsoft.com/en-us/azure/devops/pipelines/library/task-groups?view=azure-devops) کنید.

# آشنایی با بخش Library

مجموعه ای از دارایی های مرتبط با مراحل Build و Release پروژه می باشد. این دارایی ها شامل [Variable group](https://docs.microsoft.com/en-us/azure/devops/pipelines/library/variable-groups?view=azure-devops&tabs=yaml)ها و [Secure file](https://docs.microsoft.com/en-us/azure/devops/pipelines/library/secure-files?view=azure-devops)ها است.