**« Microsoft Azure DevOps (Session12)»**

**معاونت** توسعه و برند

واحد نرم‌افزار



فهـرســت

[معرفی بخش Environment 2](#_Toc73195562)

[Deployment job 2](#_Toc73195563)

[تبدیل Classic pipeline به YAML 3](#_Toc73195564)

[استفاده از Template به جای Task group در YAML 3](#_Toc73195565)

[Maintenance job 5](#_Toc73195566)

[Maintenance history 5](#_Toc73195567)



# معرفی بخش Environment

هر [Environment](https://docs.microsoft.com/en-us/azure/devops/pipelines/process/environments?view=azure-devops) مجموعه ای از منابع است که هرکدام به عنوان هدف استقرار یک Pipeline بوده و بر روی آنها فرآیند Deploy بصورت یکپارچه انجام می‌پذیرد.

این مفهوم دارای دو مسئولیت مهم می‌باشند:

1. محیط هایی که قرار است جهت اجرای فرآیند deploy مورد استفاده قرار گیرند را به تیم توسعه ارائه می‌دهد.
2. امکاناتی جهت مدیریت کانسپت Approverها در اختیار تیم توسعه می‌گذارند.

در حال حاضر دو نوع Environment قابل تعریف می‌باشد:

1. [Kubernetes resource](https://docs.microsoft.com/en-us/azure/devops/pipelines/process/environments-kubernetes?view=azure-devops)
2. [Virtual machine resource](https://docs.microsoft.com/en-us/azure/devops/pipelines/process/environments-virtual-machines?view=azure-devops)

**نکته:** این امکان فراهم است تا یک Environment خالی ایجاد کنیم و بعداً به آن منابع را اختصاص دهیم.

**نکته:** Environmentها تنها برای پایپلاین‌های YAML base مورد استفاده قرار می‌گیرند.

**نکته:** می‌توان با استفاده از تگ‌ها جهت دسته بندی کامپیوترها و منابع یک Environment اقدام نمود.  
مثلا جدا از اینکه با استفاده از خود Environmentها میتوان این فیلتر را ایجاد کرد که کدام ماشین ها کدام Pipelineها را اجرا کنند، با استفاده از تگ‌ها هم می‌توان فیلتر دیگری افزود و تعیین کرد که در یک Environment مشخص فقط ماشین‌های دارای یک تگ خاص، Pipeline را اجرا نمایند.

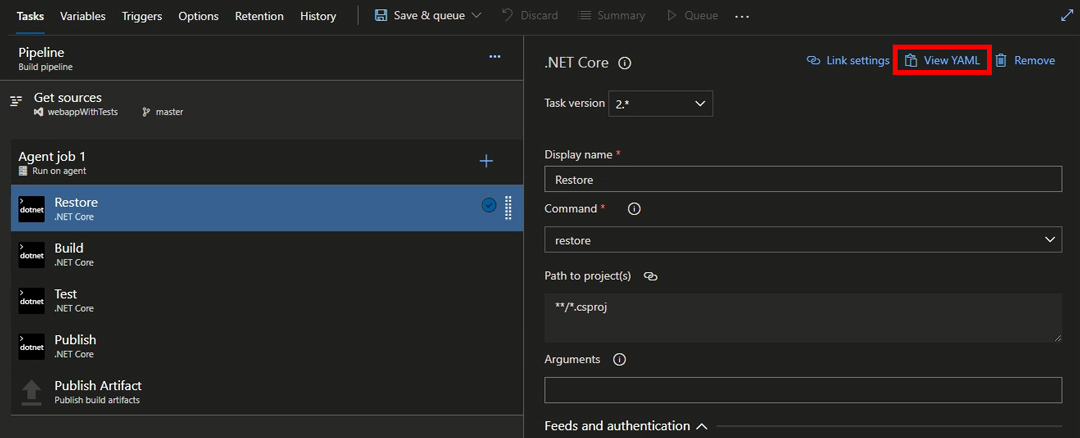
# Deployment job

در مدل YAML Pipeline توصیه می‌شوند که گام‌های deploy خود را در نوعی ویژه از job تحت عنوان deployment job قرار دهید. یک deployment job مجموعه‌ای از Stepها است که به طور متوالی در برابر Environmentها اجرا می‌شوند.

# تبدیل Classic pipeline به YAML

برای این تبدیل امکان ویژه‌ای فراهم است که هر Task قابلیت تحویل فایل YAML مرتبط با خود را دارد (مطابق با تصویر زیر).

بنابراین برای تکمیل این تبدیل باید دستورات YAML تمامی تسک ها را یک به یک دریافت کرده و به ترتیب در فایل YAML جایگذاری نمائیم. مواردی از نوع Script نیز عیناً به فایل YAML منتقل می‌شوند.



**نکته:** برای تبدیل Task groupها به دستورات YAML می‌بایست وارد آن شده و تمامی Taskها را بصورت جداگانه تبدیل نمائیم.

# استفاده از Template به جای Task group در YAML

در متد YAML این امکان فراهم است تا از دستورات خود یک الگو ایجاد نموده و به جای Task groupها که در متد Classic مورد استفاده قرار می‌گرفتند، استفاده نمائیم.  
بنابراین YAMLها یک mode پیچیده تری دارند که می‌شود templateهای گوناگون پیاده سازی کرده و هر زمان که لازم به استفاده بود آنها را فراخوانی نمود.

 تنظیمات Agent Pool

Jobهایی که توسط آن Agent pool انجام شده است را لیست می‌کند.

**Agents**

تعداد Agentهای نصب شده بر روی pool و وضعیت هرکدام را نشان می‌دهد.

Online: یعنی agent آماده به اجرا است.

Offline: یعنی agent قابل استفاده نیست.

Enabled: در صورت غیرفعال بودن امکان استفاده از agent فراهم نیست حتی اگر online باشد.

**Security**

جهت مدیریت تخصیص و استفاده از Agentها بکار می‌رود.

در دو سطح کالکشن و تیم پروژه قابل تعریف است که هر سطح دارای سه Role جداگانه می‌باشد:

در سطح کاکشن:

1. Reader

این نقش تنها امکان مانیتورینگ Agentها و بررسی صحت کارکرد آنها را فراهم می‌آورد.

1. Service Account

این نقش این امکان را فراهم می‌آورد تا اشخاص با استفاده از Agent poolهای سطح کالکشن یا سازمان، Agent poolهایی در سطح پروژه ایجاد نمایند.

1. این سطح دسترسی کامل جهت ایجاد و حذف Agentها از Pool مورد نظر را داشته و می‌توانند نقش سایر اشخاص را در این سطح تعیین و تنظیم نمایند. شخصی که Pool را ایجاد می‌کند بطور معمول در این نقش قرار می‌گیرد.

در سطح تیم پروژه:

1. Reader

امکان مشاهده Agent pool را داشته و بطور معمول جهت اضافه نمودن اشخاصی با هدف مانیتورینگ build و deployment jobs بکار می‌روند.

1. User

این نقش دسترسی استفاده از Agentها را در زمان ایجاد Pipelineها دارند.

1. Administrator

این نقش امکان مدیریت تماس اشخاص و roleهای تخصیصی به آنها را دارد. شخصی که Pool را ایجاد می‌کند بطور معمول در این نقش قرار می‌گیرد



# Maintenance job

مفهوم [Maintenance job](https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/pools-queues?view=azure-devops&tabs=yaml%2Cbrowser" \l "what-is-a-maintenance-job) در Agent pool که از تب setting قابل تعریف و شخصی سازی است، جهت پاکسازی دایرکتوری ها و ریپازیتوری های قدیمی بصورت دوره ای بکار می‌رود. این اقدام جهت مدیریت و افزایش فضای خالی دیسک انجام می‌پذیرد.  
Maintenance job در سطح کالکشن تنظیم می‌گردد.

Maintenance job timeout: حداکثر زمانی که یک Agent می‌تواند پیوسته به فعالیت Maintenance بپردازد.

Maintenance Percentage: درصدی از Agentها که بصورت همزمان می‌توانند فعالیت Maintenance را انجام دهند.

# Maintenance history

از تب Maintenance history نیز می‌توان این فعالیت را مانیتور کرده و لاگ آن را دانلود و بررسی نمود.



**Secure fileها در بخش library**

شامل فایل‌هایی می‌شوند که تیم توسعه و یا سازمان تمایلی به نگهداری آن‌ها بر روی ریپازیتوری خود نداشته اما در فرآیند Release باید مورد استفاده قرار گیرند. مانند Certificates، Android Keystore files و SSH keys

بکارگیری این فایل ها در فرآیند Release با استفاده از Task مرتبط با دانلود Secure fileها صورت می‌پذیرد.

**نکته:** هنگامی که Secure file به library اضافه می‌گردد، محتوای آن قابل رویت و ویرایش نیست و صرفا جهت فراخوانی در Release بکار می‌روند.

مطالعه بیشتر از طریق [این لینک](https://docs.microsoft.com/en-us/azure/devops/pipelines/library/secure-files?view=azure-devops#:~:text=Secure%20files%20are%20defined%20and,referencing%20them%20from%20a%20task.)