DevOps

Week 08

Murtaza Munawar Fazal

What is a Release?

- A release is a package or container containing a versioned set of artifacts specified in a release pipeline in your CI/CD process.
- It includes a snapshot of all the information required to carry out all the tasks and actions in the release pipeline, such as:
  - The stages or environments.
  - The tasks for each one.
  - The values of task parameters and variables.
  - The release policies such as triggers, approvers, and release queuing options.
- There can be multiple releases from one release pipeline (or release process).

Deployments

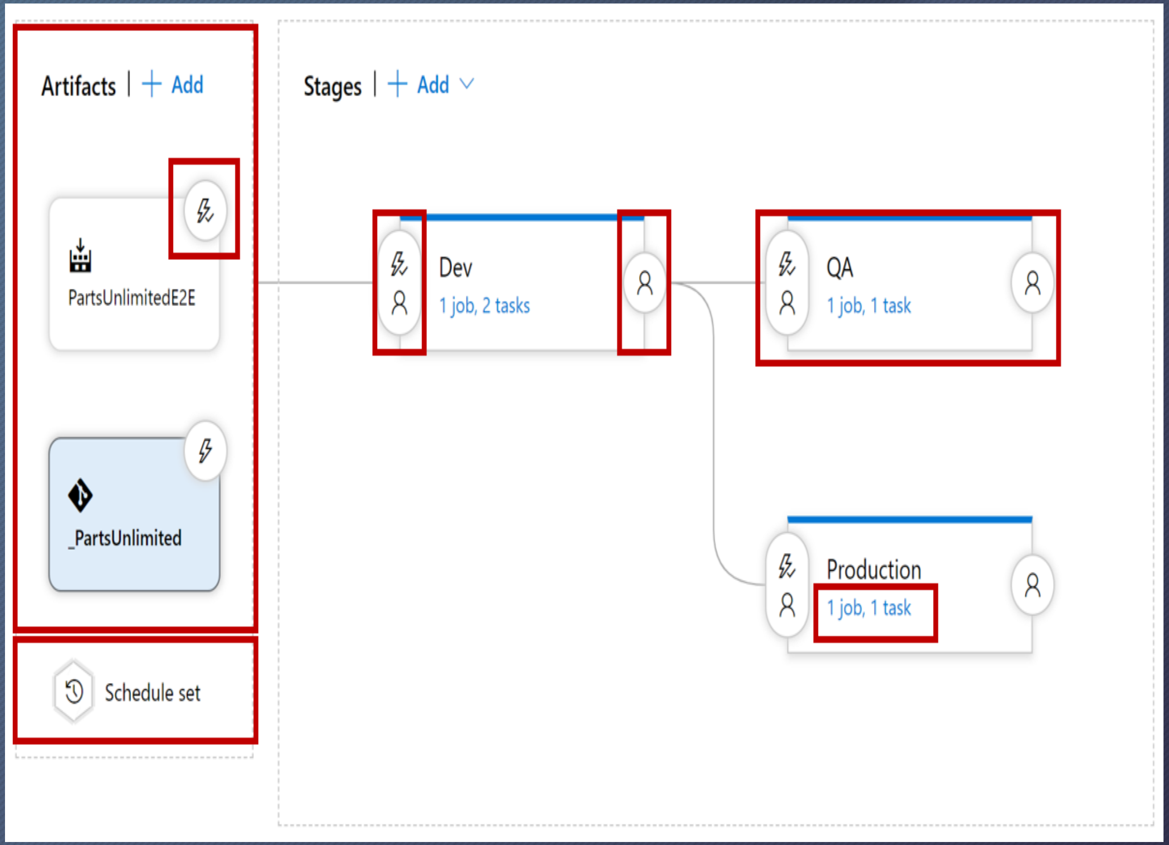
- Deployment is the action of running the tasks for one stage, which results in a tested and deployed application and other activities specified for that stage.
- Starting a release starts each deployment based on the settings and policies defined in the original release pipeline.
- There can be multiple deployments of each release, even for one stage.
- When a release deployment fails for a stage, you can redeploy the same release to that stage.

Release Process versus Release

- Release Process
  - The release pipeline (also known as release process) contains all the steps you walk through when you move your artifact from one of the artifact sources through the stages or environments.
  - The stage can be a development stage, a test stage, a production stage, or a stage where a specific user can access the application.
  - Part of your pipeline is the people who approve the release or the deployment to a specific stage. Also, triggers or schedules on which the releases execute, and the release gates, the automatic approvals of the process.
- Release
  - The release is an instance of the release pipeline. You can compare it with object instantiation.

Release Pipeline – Azure DevOps

- Components of Release Pipeline:
  - Artifacts
  - Trigger
  - Stages or Environments
  - Approval
  - Tasks



Release Pipeline – Azure DevOps

- Artifacts:
  - Artifacts can come from different sources.
    - The most common source is a package from a build pipeline.
    - Another commonly seen artifact source is, for example, source control.
- Trigger: the mechanism that starts a new release.
  - A trigger can be:
    - A manual trigger, where people start to release by hand.
    - A scheduled trigger, where a release is triggered based on a specific time.
    - A continuous deployment trigger, where another event triggers a release. For example, a completed build.

Release Pipeline – Azure DevOps

- Another component of a release pipeline is stages or sometimes called environments.

- It's where the artifact will be eventually installed.

- For example, the artifact contains the compiled website installed on the webserver or somewhere in the cloud. You can have many stages (environments); part of the release strategy is finding the appropriate combination of stages.

Release Pipeline – Azure DevOps

- Another component of a release pipeline is approval.
- People often want to sign a release before installing it in the environment.
- In more mature organizations, this manual approval process can be replaced by an automatic process that checks the quality before the components move on to the next stage.

Release Pipeline – Azure DevOps

- The tasks are the steps that need to be executed to install, configure, and validate the installed artifact.

Artifact

- An artifact is a deployable component of your application. These components can then be deployed to one or more environments.
- In general, the idea about build and release pipelines and Continuous Delivery is to build once and deploy many times.
- It means that an artifact will be deployed to multiple environments. The artifact should be a stable package if you want to achieve it.
- The configuration is the only thing you want to change when deploying an artifact to a new environment.
- The contents of the package should never change. It's what we call immutability. We should be 100% sure that the package that we build, the artifact, remains unchanged.

Build / Release Pipelines

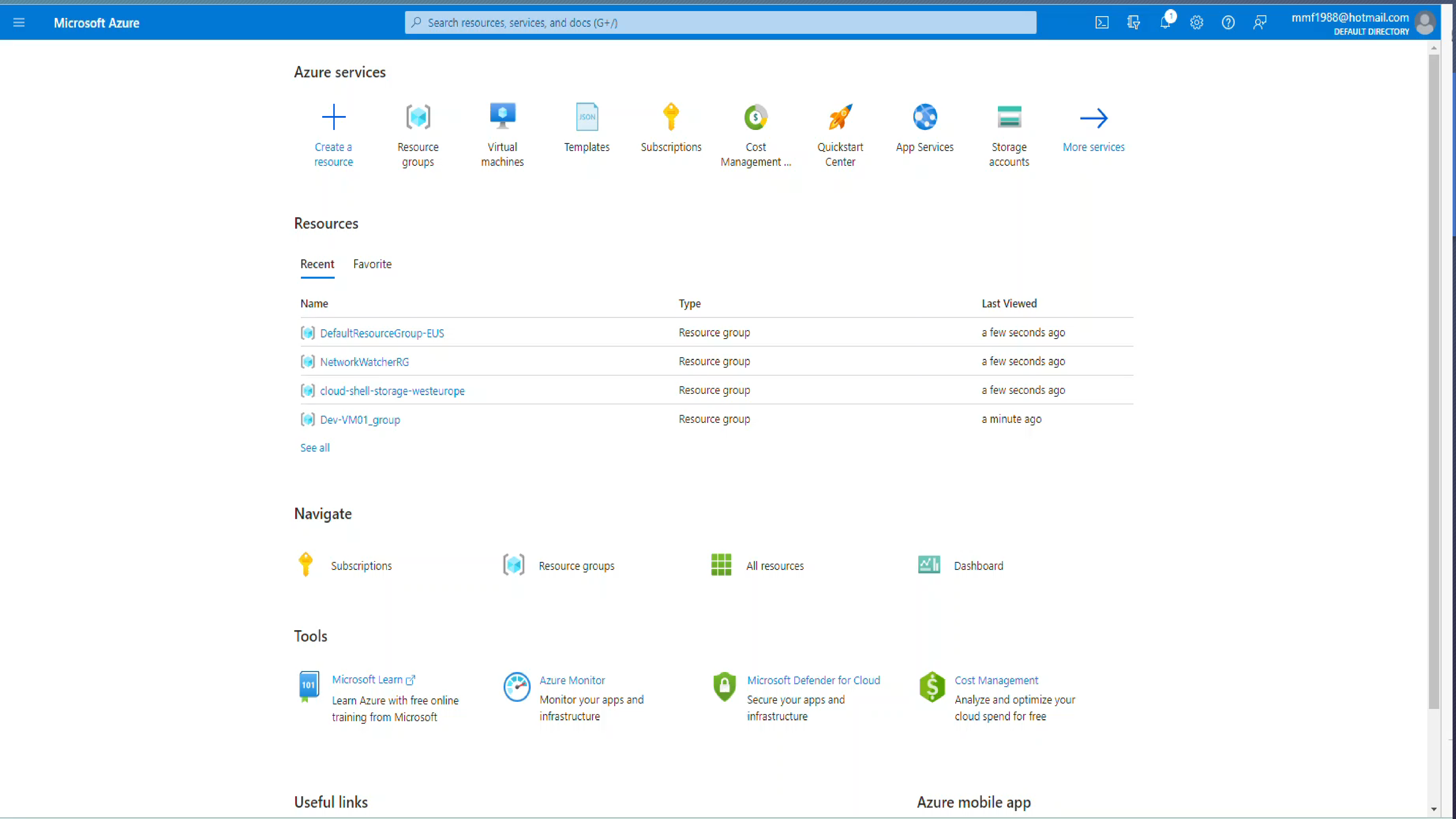
- The build pipeline compiles, tests, and eventually produces an immutable package stored in a secure place (storage account, database, and so on).
- The release pipeline then uses a secure connection to this secured place to get the build artifact and do extra actions to deploy it to an environment.
- The significant advantage of using a build artifact is that the build produces a versioned artifact.
- The artifact is linked to the build and gives us automatic traceability.
- We can directly link our version control to our release pipeline.
- The release is related to a specific commit in our version control system. With that, we can also see which version of a file or script is eventually installed. In this case, the version doesn't come from the build but from version control.
- Consideration for choosing a version control artifact instead of a build artifact can be that you only want to deploy one specific file. If you don't need to run more actions before using this file in your release pipeline, creating a versioned package (build artifact) containing only one file doesn't make sense.

Stages

- It's where the artifact will be eventually installed. For example, the artifact contains the compiled website installed on the webserver or somewhere in the cloud. You can have many stages (environments).

Demo

Configuring Pipelines as Code with YAML



Name	Type	Last Viewed
DefaultResourceGroup-EUS	Resource group	a few seconds ago
NetworkWatcherRG	Resource group	a few seconds ago
cloud-shell-storage-westeurope	Resource group	a few seconds ago
Dev-VM01_group	Resource group	a minute ago

Microsoft Learn
Learn Azure with free online training from Microsoft

Azure Monitor
Monitor your apps and infrastructure

Microsoft Defender for Cloud
Secure your apps and infrastructure

Cost Management
Analyze and optimize your cloud spend for free

Release Approvals

- Continuous Delivery is all about delivering on-demand. But it's only the technical part of the Continuous Delivery process.
- It's all about how you can technically install the software on an environment, but it doesn't say anything about the process that needs to be in place for a release.
- Release approvals don't control how but control if you want to deliver multiple times a day.
- Manual approvals also suit a significant need. Organizations that start with Continuous Delivery often lack a certain amount of trust.
- They don't dare to release without manual approval. After a while, when they find that the approval doesn't add value and the release always succeeds, the manual approval is often replaced by an automatic check.

Release Approvals

- While DevOps is all about automation, manual approvals are still helpful. There are many scenarios where they're needed. For example, a product owner might want to sign out a release before it moves to production.

- Or the scrum team wants to make sure that no new software is deployed to the test environment before someone signs off on it because they might need to find an appropriate time if it's constantly in use.

Release Gates

- Release gates give you more control over the start and completion of the deployment pipeline.
- They're often set up as pre-deployment and post-deployment conditions.
- In many organizations, there are so-called dependency meetings.
- Think of downtime of a database server or an update of an API.
- It takes much time and effort, and the only thing needed is a signal if the release can continue.

Release Gates

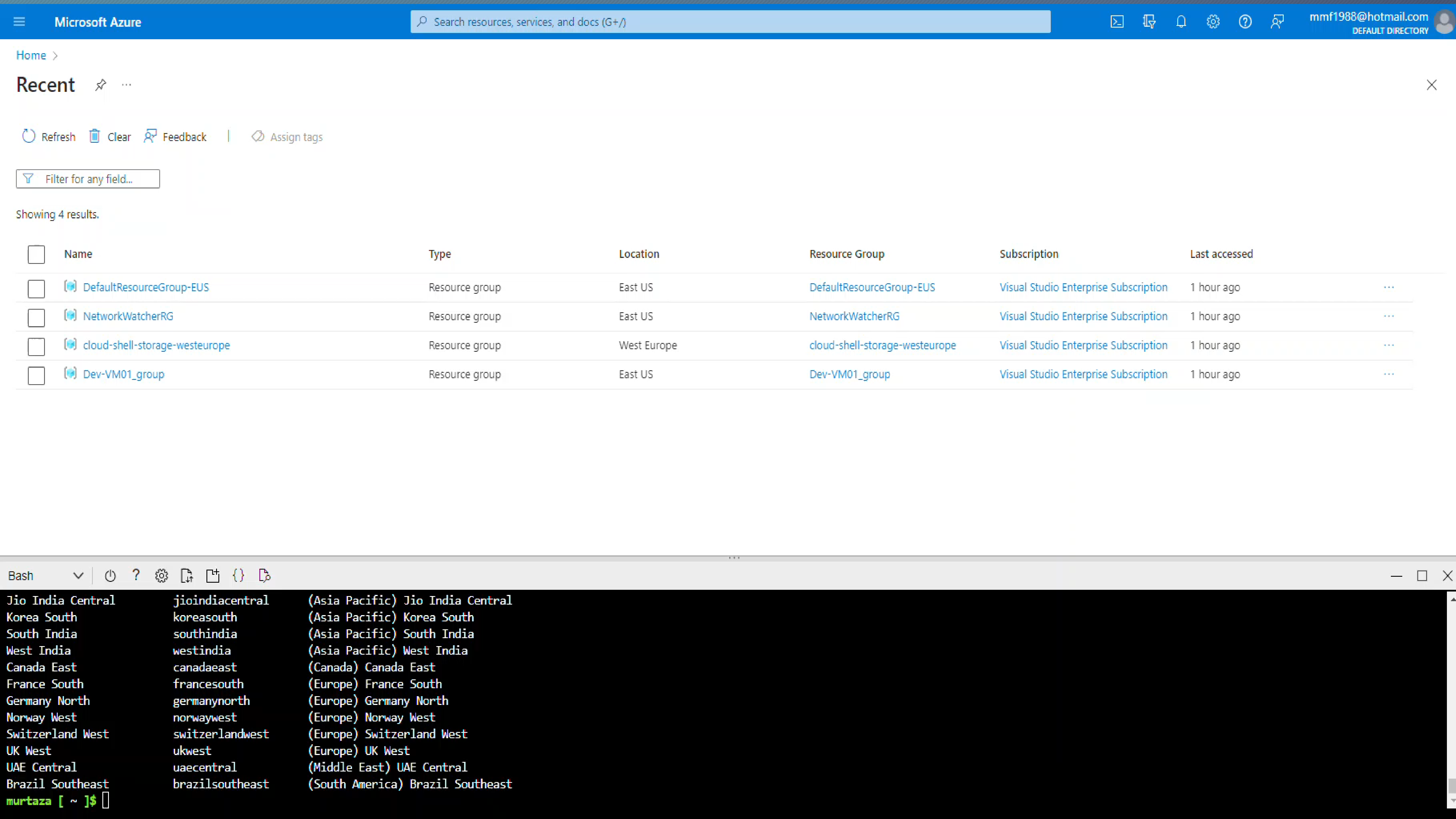
- Instead of having a meeting, you can create a mechanism where people press a button on a form when the release can't advance.
- When the release starts, it checks the state of the gate by calling an API. If the "gate" is open, we can continue. Otherwise, we'll stop the release.
- By using scripts and APIs, you can create your release gates instead of manual approval. Or at least extending your manual approval.

Release Gates - Examples

- Incident and issues management. For example, ensure that deployment only occurs if no bugs exist.
- Notify users such as legal approval departments, auditors, or IT managers about a deployment by integrating with approval collaboration systems such as Microsoft Teams or Slack and waiting for the approval to complete.
- Quality validation. Query metrics from tests on the build artifacts such as pass rate or code coverage and only deploy within required thresholds.
- Security scan on artifacts. Ensure security scans such as anti-virus checking, code signing, and policy checking for build artifacts have been completed.
- User experience relative to baseline. Using product telemetry, ensure the user experience hasn't regressed from the baseline state.
- Change management. Wait for change management procedures in a system such as ServiceNow complete before the deployment occurs.
- Infrastructure health. Execute monitoring and validate the infrastructure against compliance rules after deployment or wait for proper resource use and a positive security report.

Demo

Controlling Deployments using Classic Release Gates



Recent

Showing 4 results.

□	Name	Type	Location	Resource Group	Subscription	Last accessed
□	DefaultResourceGroup-EUS	Resource group	East US	DefaultResourceGroup-EUS	Visual Studio Enterprise Subscription	1 hour ago
□	NetworkWatcherRG	Resource group	East US	NetworkWatcherRG	Visual Studio Enterprise Subscription	1 hour ago
□	cloud-shell-storage-westeurope	Resource group	West Europe	cloud-shell-storage-westeurope	Visual Studio Enterprise Subscription	1 hour ago
□	Dev-VM01_group	Resource group	East US	Dev-VM01_group	Visual Studio Enterprise Subscription	1 hour ago

```
Jio India Central          jioindiacentral       (Asia Pacific) Jio India Central
Korea South                koreasouth            (Asia Pacific) Korea South
South India                southindia            (Asia Pacific) South India
West India                 westindia             (Asia Pacific) West India
Canada East                canadaeast            (Canada) Canada East
France South               francesouth           (Europe) France South
Germany North              germanynorth          (Europe) Germany North
Norway West                norwaywest            (Europe) Norway West
Switzerland West           switzerlandwest       (Europe) Switzerland West
UK West                    ukwest                (Europe) UK West
UAE Central                uaecentral            (Middle East) UAE Central
Brazil Southeast           brazilsoutheast       (South America) Brazil Southeast
murtaza [ ~ ]$
```