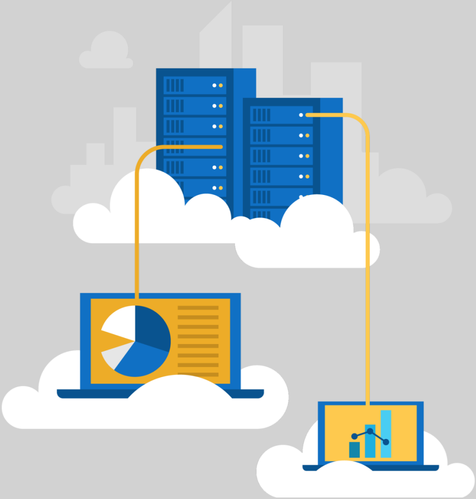
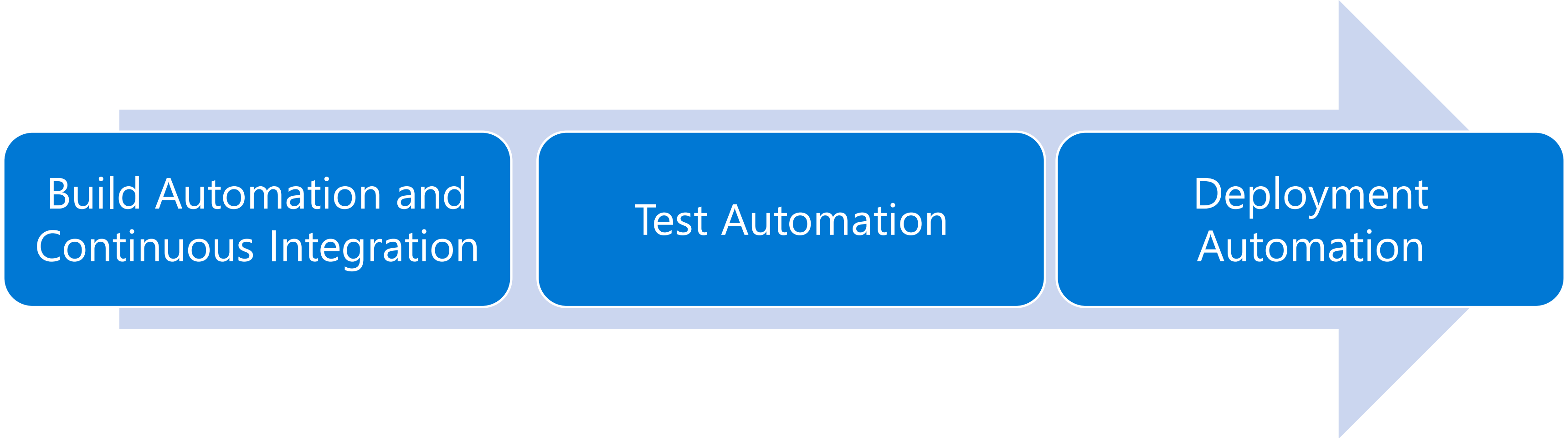
Microsoft

Module 3: Implement and Manage Build Infrastructure

Lesson 01: The Concept of Pipelines in DevOps



The Concept of Pipelines in DevOps


Build Automation and Continuous Integration

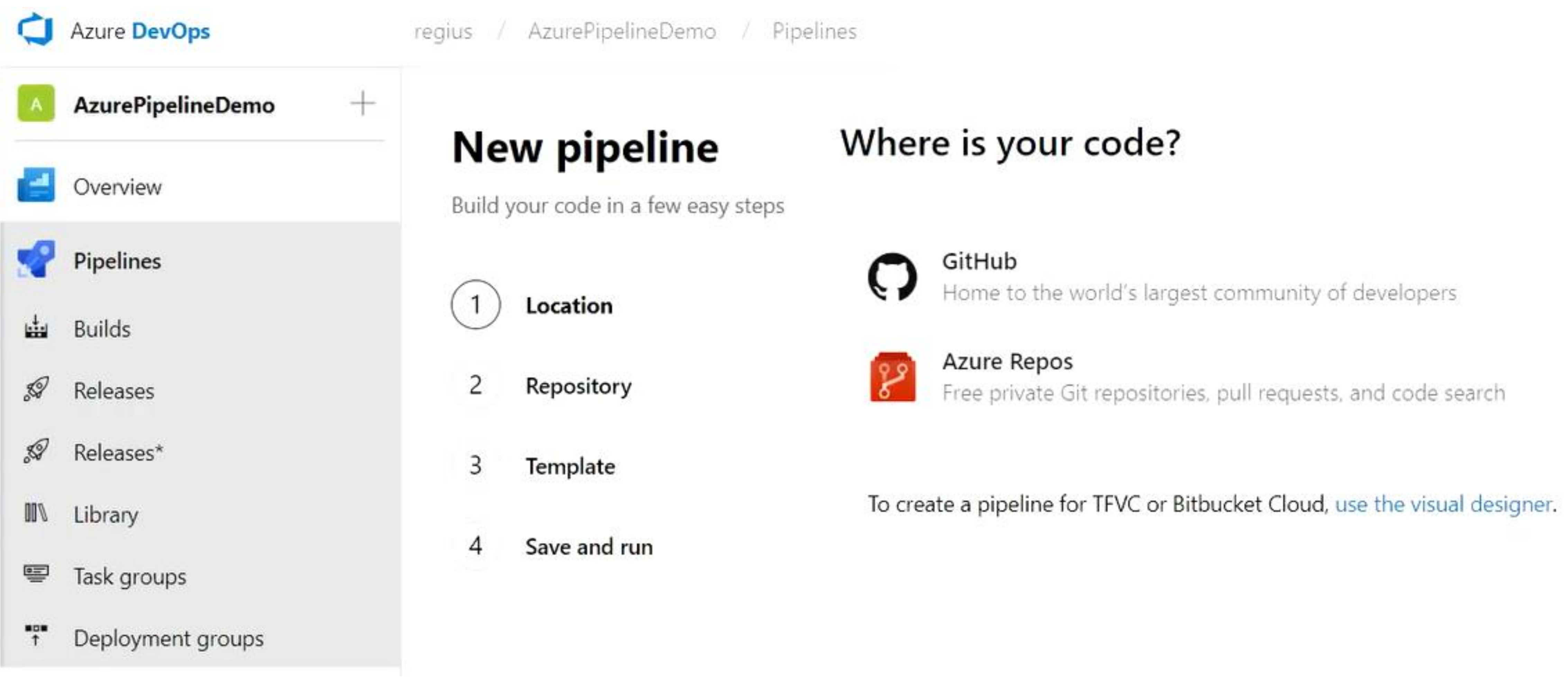
Test Automation

Deployment Automation

- A pipeline enables a constant flow of changes into production via an automated software production line
- Pipelines create a repeatable, reliable and incrementally improving process for taking software from concept to customer
- Pipelines require infrastructure, this infrastructure will have a direct impact on the effectiveness of the pipeline

Lesson 02: Azure Pipelines

Azure Pipelines



New pipeline

Build your code in a few easy steps

1 Location

2 Repository

3 Template

4 Save and run

Where is your code?

GitHub
Home to the world's largest community of developers

Azure Repos
Free private Git repositories, pull requests, and code search

To create a pipeline for TFVC or Bitbucket Cloud, use the visual designer.

Azure Pipelines

- Azure Pipelines is a cloud service that you can use to automatically build and test your code project and make it available to other users
- Works great with Continuous Integration and Continuous Delivery
  - Work with any language or platform - Python, Java, PHP, Ruby, C#, and Go
  - Deploy to different types of targets at the same time
  - Integrate with Azure deployments - container registries, virtual machines, Azure services, or any on-premises or cloud target (Microsoft Azure, Google Cloud, or Amazon cloud services)
  - Build on Windows, Linux, or Mac machines
  - Integrate with GitHub
  - Work with open-source projects

Azure Key Terms

- Agent
- Artifact
- Build
- Continuous Delivery
- Continuous Integration
- Deployment Target
- Job
- Pipeline
- Release
- Task
- Trigger

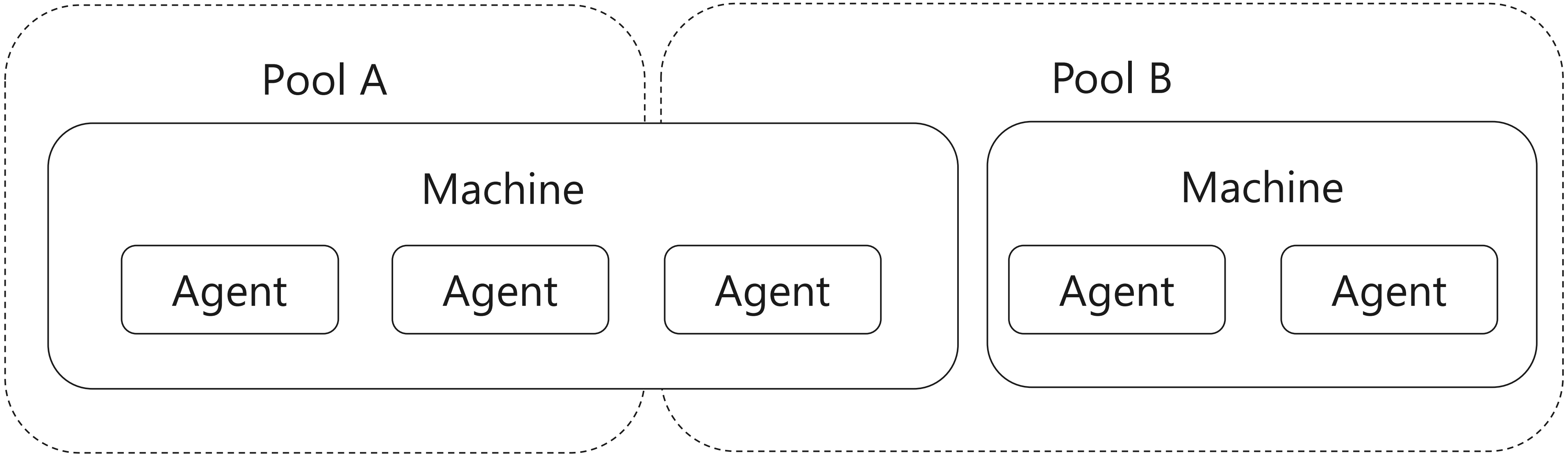
Lesson 03: Evaluate Use of Hosted vs Private Agents

Hosted vs Private Agents

- You need at least one agent to build or deploy your project
- An agent is installable software that runs one build or deployment job at a time
- Two types of agents:
  - **Microsoft-hosted agents** - Automatically take care of maintenance and upgrades. Each time you run a pipeline, you get a fresh virtual machine. The virtual machine is discarded after one use.
  - **Self-hosted agents** – You take care of maintenance and upgrades. Give you more control to install dependent software needed. You can install the agent on Linux, macOS, Windows machines, or even in a Linux Docker container.

Lesson 04: Agent Pools

Agent Pools



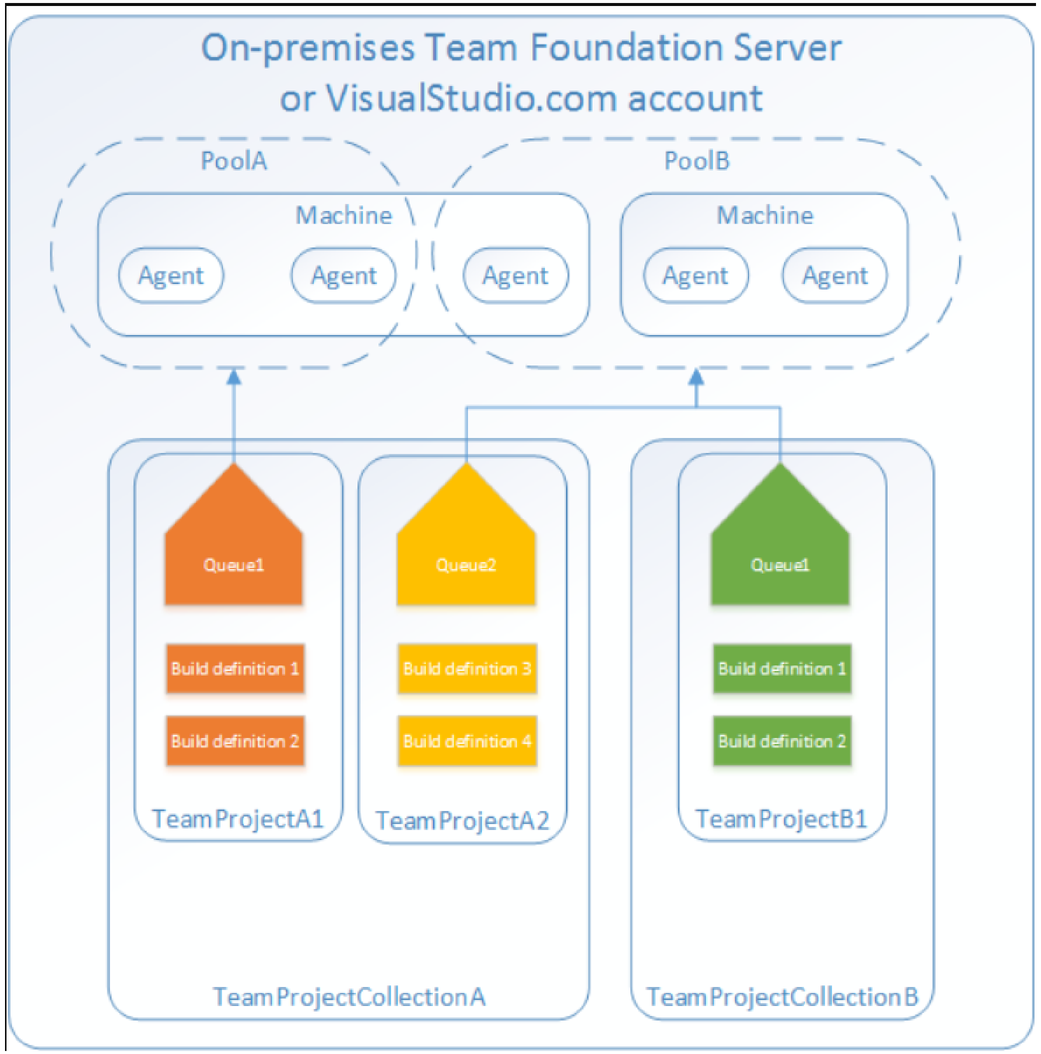
- You can organize agents into agent pools
- An agent pool defines the sharing boundary
- In Azure Pipelines, agent pools are scoped to the Azure DevOps organization; so you can share an agent pool across projects

Default Agent Pools

- Default pool – For self-hosted agents
- Hosted Ubuntu 1604 pool
- Hosted macOS pool
- Hosted VS2017 pool
- Hosted pool – Microsoft pool
- Hosted Windows Container pool

Typical Situations for Agent Pools

- You're a member of a project and you want to use a set of machines owned by your team for running build and deployment jobs
- You're a member of the infrastructure team and would like to set up a pool of agents for use in all projects
- You want to share a set of agent machines with multiple projects, but not all of them



Security of Agent Pools

- Roles are defined on each agent pool, and membership in these roles governs what operations you can perform on an agent pool

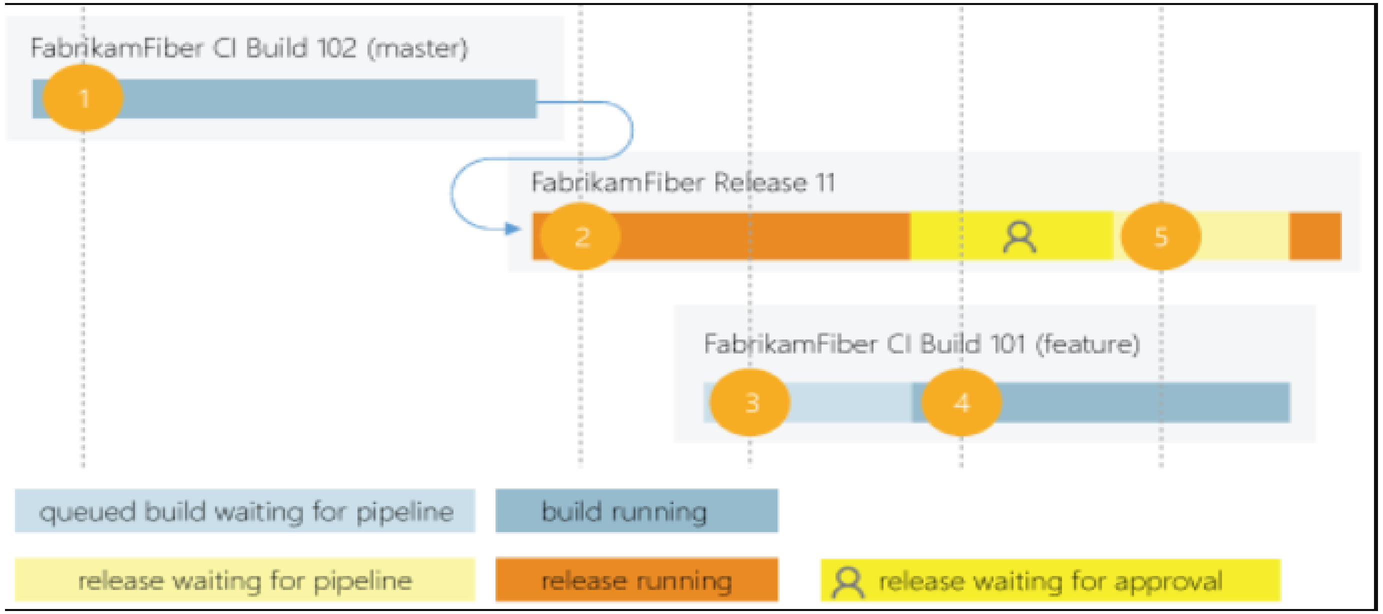
Role	Purpose
Reader	Members of this role can view the organization agent pool as well as agents. You typically use this to add operators that are responsible for monitoring the agents and their health.
Service Account	Members of this role can use the organization agent pool to create a project agent pool in a project. If you follow the guidelines above for creating new project agent pools, you typically do not have to add any members here.
Administrator	In addition to all the above permissions, members of this role can register or unregister agents from the organization agent pool. They can also refer to the organization agent pool when creating a project agent pool in a project. Finally, they can also manage membership for all roles of the organization agent pool. The user that created the organization agent pool is automatically added to the Administrator role for that pool.

Lesson 05: Pipelines and Concurrency

Microsoft-Hosted vs Self-Hosted

- A free tier of service by default for every organization
- Service is based on public vs private projects
- Microsoft-hosted CI/CD
  - Run your builds and releases on machines that Microsoft manages
  - Jobs run on the pool of hosted agents
  - There are time limits on these jobs
- Self-hosted CI/CD
  - Azure Pipelines orchestrates your builds and releases
  - Use your own machines to run the builds and releases
  - There are no time limits on these jobs

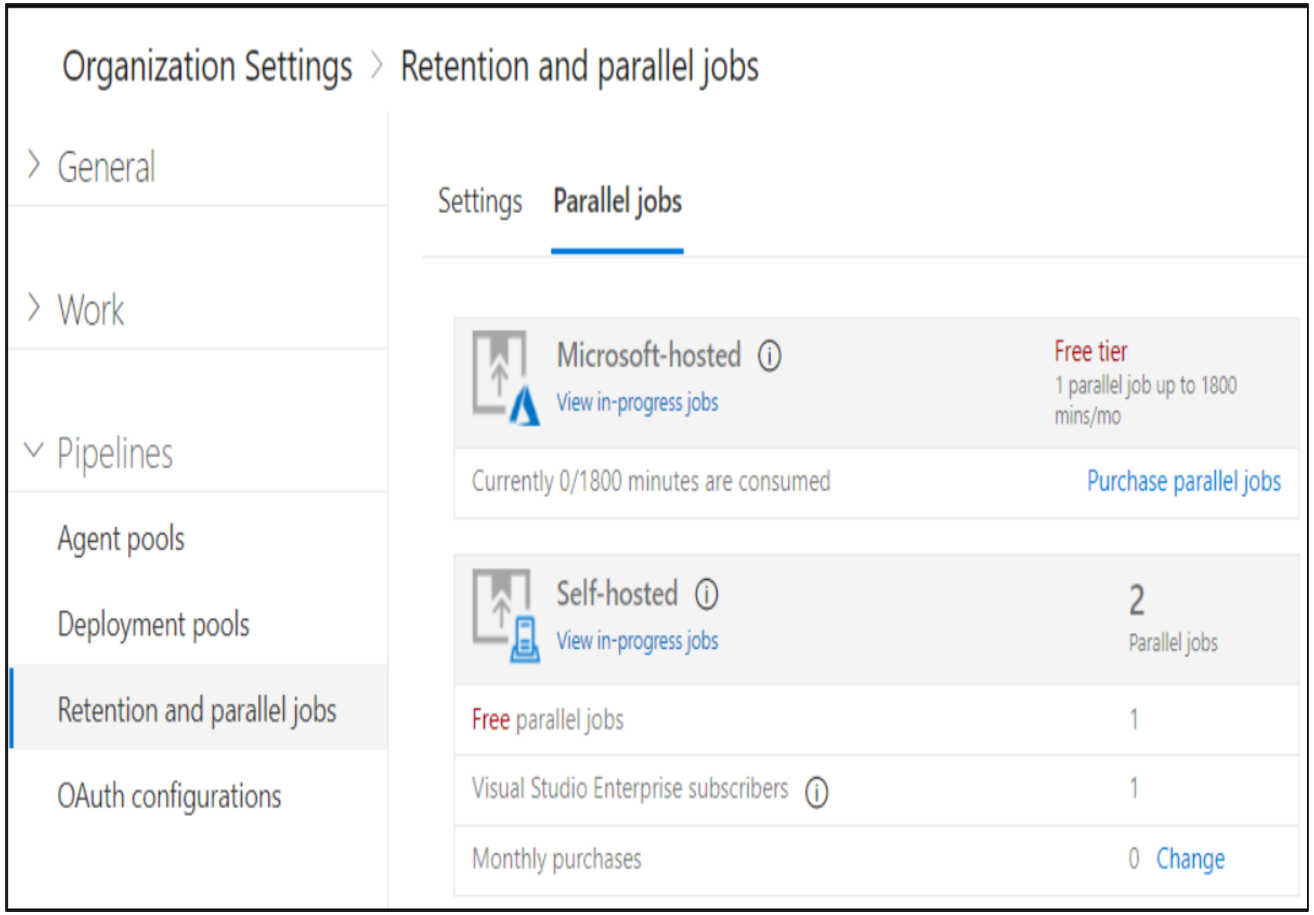
Parallel Jobs


FabrikamFiber CI Build 102 (master)
1
FabrikamFiber Release 11
2
5
FabrikamFiber CI Build 101 (feature)
3
4
queued build waiting for pipeline
build running
release waiting for pipeline
release running
release waiting for approval

- This is a simple example of a Microsoft-hosted parallel job
- Users can collectively run only one build or release job at a time, additional jobs are queued
- A release consumes a parallel job only when it's being actively deployed to a stage

Estimating Parallel Jobs

- Determine how many parallel jobs you need
- Simple estimates vs Detailed estimates
- You can display all the builds and releases
- Parallel jobs are purchased at the organization level, and are shared by all projects



Microsoft-hosted ⓘ

Free tier
1 parallel job up to 1800 mins/mo

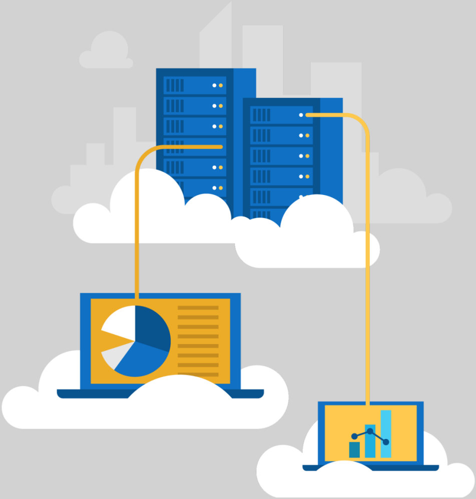
Currently 0/1800 minutes are consumed

Self-hosted ⓘ

2
Parallel jobs

Free parallel jobs	1
Visual Studio Enterprise subscribers ⓘ	1
Monthly purchases	0

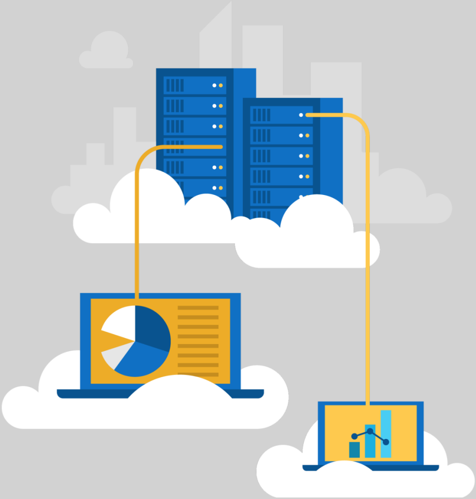
Lesson 06: Azure DevOps and Open Source Projects



Azure DevOps and Open Source Projects

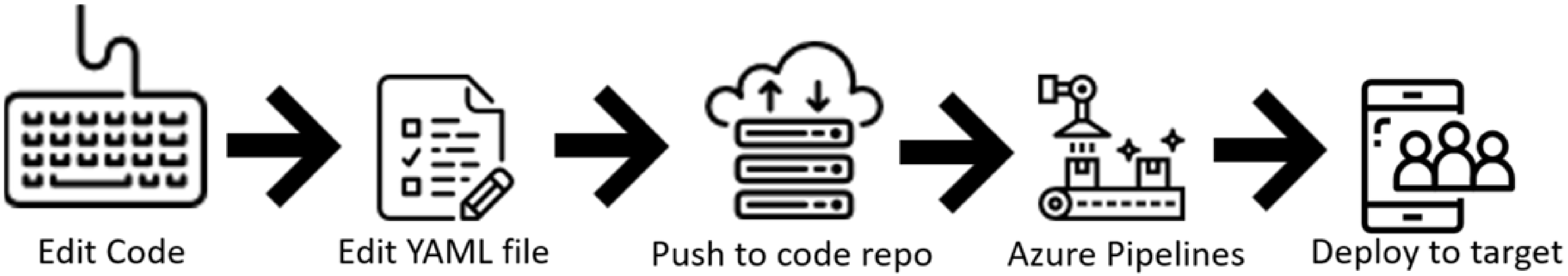
- How do I qualify for the free tier of public projects?
- Are there limits on who can use Azure Pipelines?
- Are there any limits on the number of builds and release pipelines that I can create?
- As a Visual Studio Enterprise subscriber, do I get additional parallel jobs for TFS and Azure Pipelines?
- What about the option to pay for hosted agents by the minute?

Lesson 07: Azure Pipelines YAML vs Visual Designer



Azure Pipelines and YAML

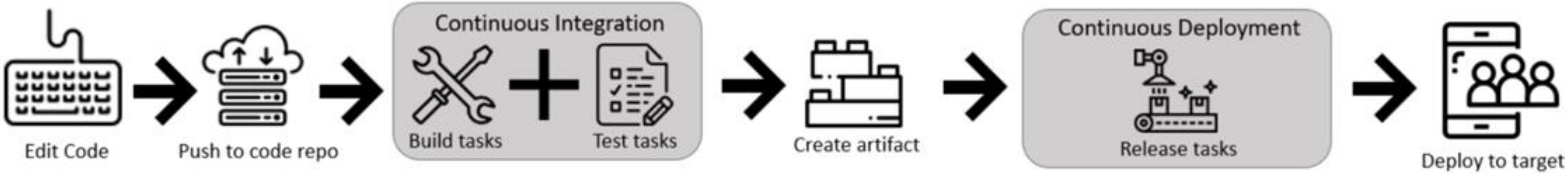
- Configure your pipelines in a YAML file that exists alongside your code



- Configure Azure Pipelines to use your Git repo
- Edit your azure-pipelines.yml file to define your build
- Push your code to your version control repository
- Your code is now updated, built, tested, and packaged

Azure Pipelines and Visual Designer

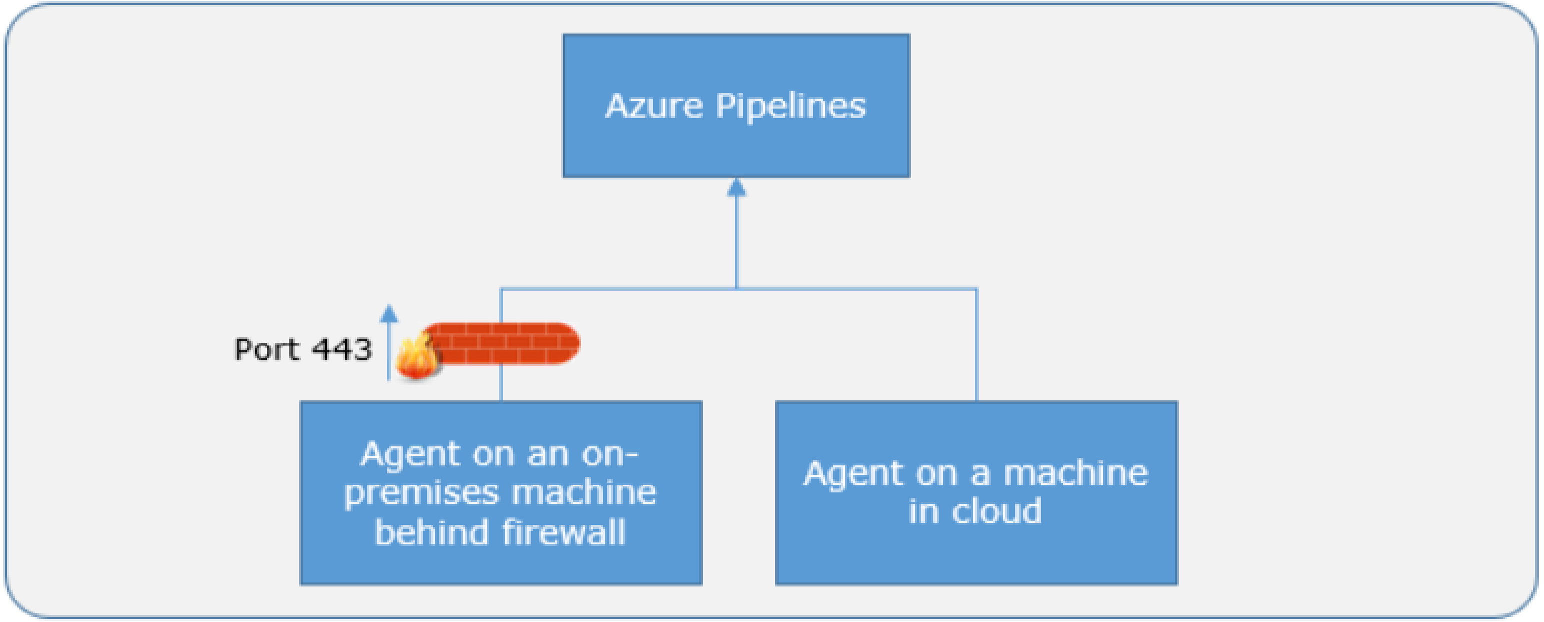
- Configure your pipelines with the Visual Designer



- Create and configure your build and release pipelines
- Push your code to your version control repository
- The build creates an artifact that's used by the rest of your pipeline
- Your code is now updated, built, tested, and packaged

Lesson 08: Setup Private Agents

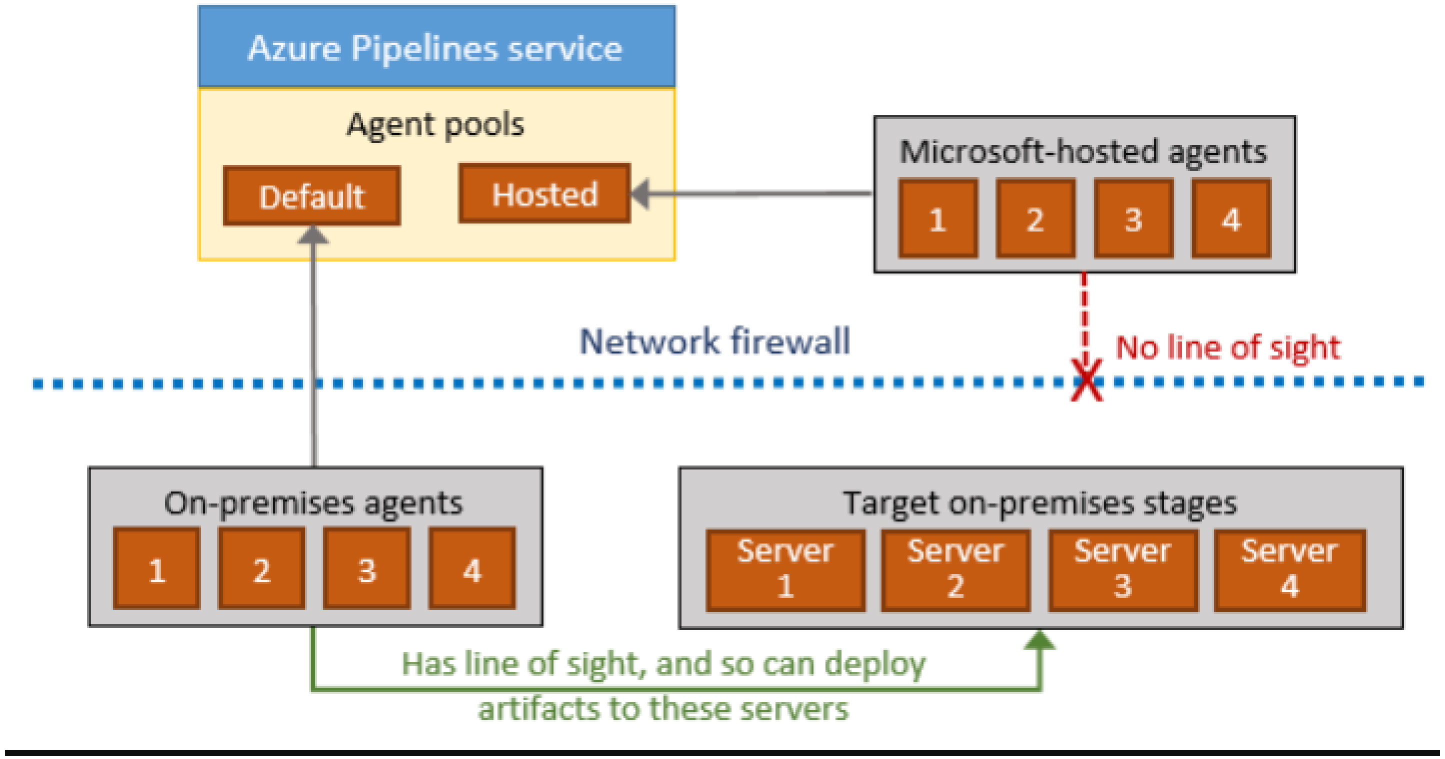
Communication with Azure Pipelines


Azure Pipelines
Port 443
Agent on an on-premises machine behind firewall
Agent on a machine in cloud

- The agent determines which job it needs to run, and to report the logs and job status
- Communication is always initiated by the agent
- All the messages from the agent to Azure Pipelines are over HTTPS

Communication to Deploy to Target Servers

- Agent must have "line of sight" connectivity to servers
- Microsoft-hosted agent pools, by default, have connectivity to Azure websites and servers running in Azure
- You may need to manually configure connectivity



Other Considerations

- Authentication
- Personal Access Tokens
- Interactive vs Service processes
- Agent version and upgrades

Lesson 09: Integrate Jenkins with Azure Pipelines

Lab: Configuring a CD pipeline for your Jenkins CI

In this lab, Configuring a CD pipeline for your Jenkins CI, you will examine two methods for integrating Jenkins: run CI jobs in Jenkins separately, and wrap a Jenkins CI job inside an Azure pipeline. You will learn how to:

- Provision Jenkins on Azure VM using the Jenkins template available on the Azure Marketplace
- Configure Jenkins to work with Maven and Azure DevOps
- Create a build job in Jenkins
- Configure Azure Pipeline to integrate with Jenkins
- Configure a CD pipeline in Azure Pipelines to deploy the build artifacts

✓ Note that you must have already completed the prerequisite labs in the Welcome section.

Lesson 10: Integration External Source Control with Azure Pipelines

Lab: Integrate Your GitHub Projects With Pipelines

In this lab, Integrate Your GitHub Projects With Azure Pipelines, you will how easy it is to set up Azure Pipelines with your GitHub projects and how you can start seeing benefits immediately. You will learn how to:

- Install Azure Pipelines from the GitHub Marketplace.
- Integrate a GitHub project with an Azure DevOps pipeline.
- Track pull requests through the pipeline.

✓ Note that you must have already completed the prerequisite labs in the Welcome section.

Lesson 11: Analyze and Integrate Docker Multi-Stage Builds

Builder Patterns

- Derive from a dotnet base image with the whole runtime/SDK (Dockerfile.build)
- Add source code
- Produce a statically-linked binary
- Copy the static binary from the image to the host (docker create, docker cp)
- Derive from SCRATCH or some other light-weight image (Dockerfile)
- Add the binary back in
- Push a tiny image to the Docker Hub

Multi-stage Builds

- Multi-stage builds give the benefits of the builder pattern without the hassle of maintaining separate files
- The code progresses from one stage to the next

```
FROM microsoft/aspnetcore:2.0 AS base
WORKDIR /app
EXPOSE 80

FROM microsoft/aspnetcore-build:2.0 AS builder
WORKDIR /src
COPY *.sln ./
COPY Web/Web.csproj Web/
RUN dotnet restore
COPY . .
WORKDIR /src/Web
RUN dotnet build -c Release -o /app

FROM builder AS publish
RUN dotnet publish -c Release -o /app

FROM base AS production
WORKDIR /app
COPY --from=publish /app .
ENTRYPOINT ["dotnet", "Web.dll"]
```

Multiple Projects and Solutions

- Each project has its own multi-stage dockerfile
- Shared component projects do not have dockerfiles
- Each dockerfile assumes its context is the solution directory
- There's a docker-compose.yml in the root of the solution

```
Multi.sln
docker-compose.yml
[Api]
  Dockerfile
[Web]
  Dockerfile
```

Lab: Deploying a Multi-container Application to AKS

In this lab, Deploying a multi-container application to Azure Kubernetes Services, you will deploy and manage Docker containers using Kubernetes.

You will learn how to:

- Create an Azure Container Registry (ACR), AKS and Azure SQL server
- Provision the Azure DevOps Team Project with a .NET Core application using the Azure DevOps Demo Generator tool
- Configure application and database deployment, using Continuous Deployment (CD) in the Azure DevOps
- Initiate the build to automatically deploy the application

✓ Note that you must have already completed the prerequisite labs in the Welcome section.

Lesson 12: Module Review Questions

Module 3 Review Questions

1. What is a pipeline and why is it used?
2. What are some advantages of Azure pipelines?
3. What are the two types of agents and how are they different?
4. What is an agent pool and why would you use it?
5. What are two ways to configure your Azure pipelines?