

Microsoft Hosted vs. Self-hosted Agents

Overview



Understanding Microsoft Hosted Agents

Understanding Self-hosted Agents

Working with Agent Capabilities

Microsoft Hosted Agents

Understanding Microsoft-hosted Agents

Single-use virtual machines provided by Microsoft

All patches and upgrades are taken care of

Pre-defined software packages installed

Tasks run with the highest level of permissions

Data does not persist between pipeline runs

Additional packages can be installed



Available Hosted Agents

- Visual Studio 2019 Preview on Windows Server 2019 ([windows-2019](#))
- Visual Studio 2017 on Windows Server 2016 ([vs2017-win2016](#))
- Visual Studio 2015 on Windows Server 2012R2 ([vs2015-win2012r2](#))
- Windows Server 1803 ([win1803](#))
- macOS X Mojave 10.14 ([macOS-10.14](#))
- macOS X High Sierra 10.13 ([macOS-10.13](#))
- Ubuntu 16.04 ([ubuntu-16.04](#))



Self-hosted Agents

Understanding Self-hosted Agents

Provides much greater control over application binaries

Data caches and configuration persist between runs

Can be run on macOS, Linux, Windows and Docker

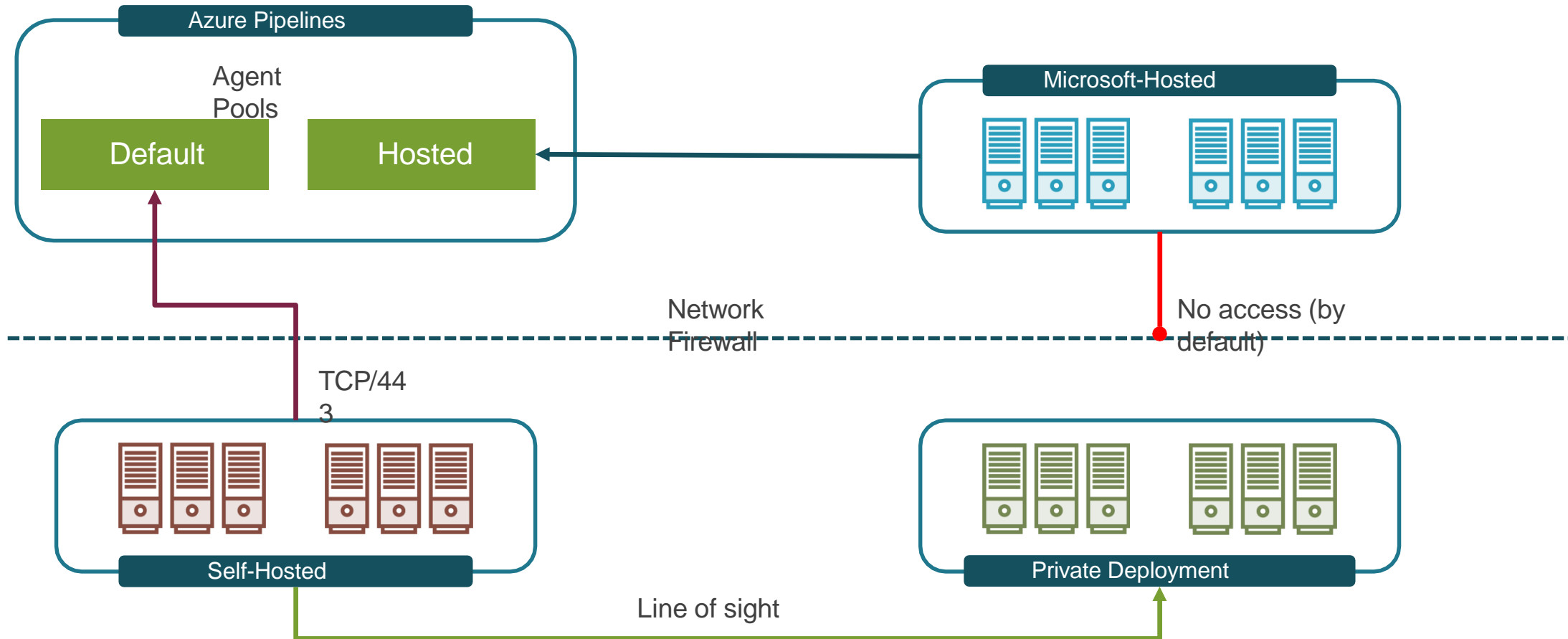
Agent runs either interactively or as a service

User is responsible for all management and configuration

User is responsible for all major version agent upgrades



Networking Considerations



<https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/agents#communication>

Working with Agent Capabilities

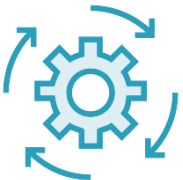
Agent Capabilities



System capabilities (automatic) and User capabilities (manual)



Includes operating system, applications and environment variables



Capabilities are used as demands in Pipeline jobs



<https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/agents#capabilities>

Agent detects system capabilities

Capability name maps to job demand

Capability names are automatically assigned

Make use of User capabilities for custom naming

System capabilities

Name	Value
Agent.Name	vsts-01
Agent.Version	2.144.0
-	./externals/node/bin/node
Agent.ComputerName	vsts-01
Agent.HomeDirectory	/home/jbannan/myagent
Agent.OS	Linux
Agent.OSArchitecture	X64
ant	/usr/bin/ant
curl	/usr/bin/curl
docker	/usr/bin/docker
git	/usr/bin/git
HOME	/home/jbannan
InteractiveSession	False

Get Agent Capabilities from REST API

\$org = '<ORG_NAME>'

\$poolId = '<AGENT_POOL_ID>'

\$agentId = '<AGENT_ID>'

\$orgUrl = [https://\\$org.visualstudio.com/](https://$org.visualstudio.com/) #Or [https://dev.azure.com/\\$org](https://dev.azure.com/$org)

\$personalToken = '<ACCESS_TOKEN>'

\$token = [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":\$((\$personalToken))"))

\$header = @{authorization = "Basic \$token"}

\$uri = "\$(\$orgUrl)_apis/distributedtask/pools/\$poolId/agents/\$agentId/?includeCapabilities=true"

Invoke-RestMethod -Uri \$uri -Method Get -ContentType "application/json" -Headers \$header

Getting Started with the REST API



Azure DevOps Services REST API Reference

[*https://docs.microsoft.com/en-us/rest/api/azure/devops*](https://docs.microsoft.com/en-us/rest/api/azure/devops)



Forming URLs to query Azure DevOps Services

[*https://docs.microsoft.com/en-us/azure/devops/extend/develop/work-with-urls*](https://docs.microsoft.com/en-us/azure/devops/extend/develop/work-with-urls)



Working with Agent Pools

[*https://docs.microsoft.com/en-us/rest/api/azure/devops/distributedtask/pools*](https://docs.microsoft.com/en-us/rest/api/azure/devops/distributedtask/pools)

Implementing Self-hosted Agents

Overview



Requirements for Self-hosted Agents

Onboard and test a Windows Agent

Onboard and test a Linux Agent

Requirements for Self-hosted Agents

Preparing for Self-hosted Agents



Check the **operating system and runtime dependencies**

<https://github.com/microsoft/azure-pipelines-agent/tree/master/docs/start>



Identify a user with **permissions to administer the agent pool**



Generate a **Personal Access Token (PAT)**



<https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/agents>

Preparing for Self-hosted Agents



HTTPS

Ensure the system can communicate outbound via HTTPS (TCP port 443)



Determine whether the agent will need to communicate via a proxy



Identify a local system account for the agent to use



<https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/agents>

Verify environment readiness

Demo



Explore Microsoft Hosted agents

Run Pipeline using Hosted agents

Demo



Explore Self-hosted agents

Run Pipeline using Self-hosted agents

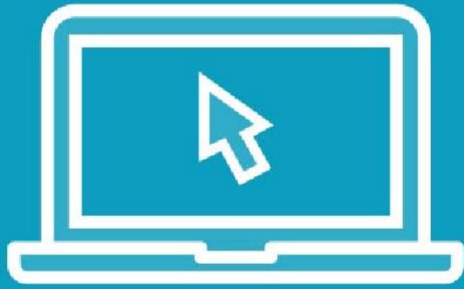
Demo



Explore agent capabilities

Use capabilities to assert job demands

Demo



Onboard a Self-hosted Windows agent

Verify Windows agent functionality

Demo



Onboard a Self-hosted Linux agent

Verify Linux agent functionality