TIP SHEET

4 Ways to Deploy to Azure Faster

The build status of your project reveals what state it's in. In order to have a successful design, it is critical that you have a build process established from the moment you get your project up and running. Identifying potential issues and ensuring that you have the most up to date information about your initiative means that having fast builds is crucial. Additionally, having a broken build can take a lot of effort to fix and the longer a build takes, the more difficult it makes deployment.

The type of builds your organization requires is contingent on many things, such as the experience of both your team and your organization, what sort of product you are creating, and your deployment approach. **There are 4 key ways to ensure that you have a fast build time:**

1 Choosing the Appropriate Build Machine

High-speed machines can change your deployment from hours to minutes. If your organization uses Azure DevOps for your pipelines, you have the option to operate using a Microsoft-hosted agent, taking care of both maintenance and enhancements.

2 Build Server Positioning

Creating code requires a lot of data moving across various platforms. Typically, information used to generate builds is gathered from both a source control repository and an artifact repository. Once the build is complete, the output needs to be replicated and reviewed, including recorded artifacts, test reports, debugging signs, and more. Ensuring that the build server is positioned near the sources and a target location, and that this is done quickly, decreasing the overall build time.

3 Build Server Scoping

Depending on the size of your product and the amount of people working on it, you may need to scale your infrastructure over several machines. One way to do this is through the use of agent pools. Rather than overseeing each agent in your pipeline, you can assign it to a specific pool and have it meet the demands of that particular pipeline, you can also share agent machines across projects.

4 Improving Your Build

One way to optimize your build is through the use of parallel jobs. If you define your pipeline as a group of jobs, you can run numerous jobs on one pipeline. With Azure Pipelines, you can execute parallel jobs without needing to pay for parallel jobs if you're using an on-premises server. Another way to improve your build is run testing through parallel jobs to improve pipeline efficiency.