

January 23, 2020

Big Car Manufacturer

A Software Development Platform



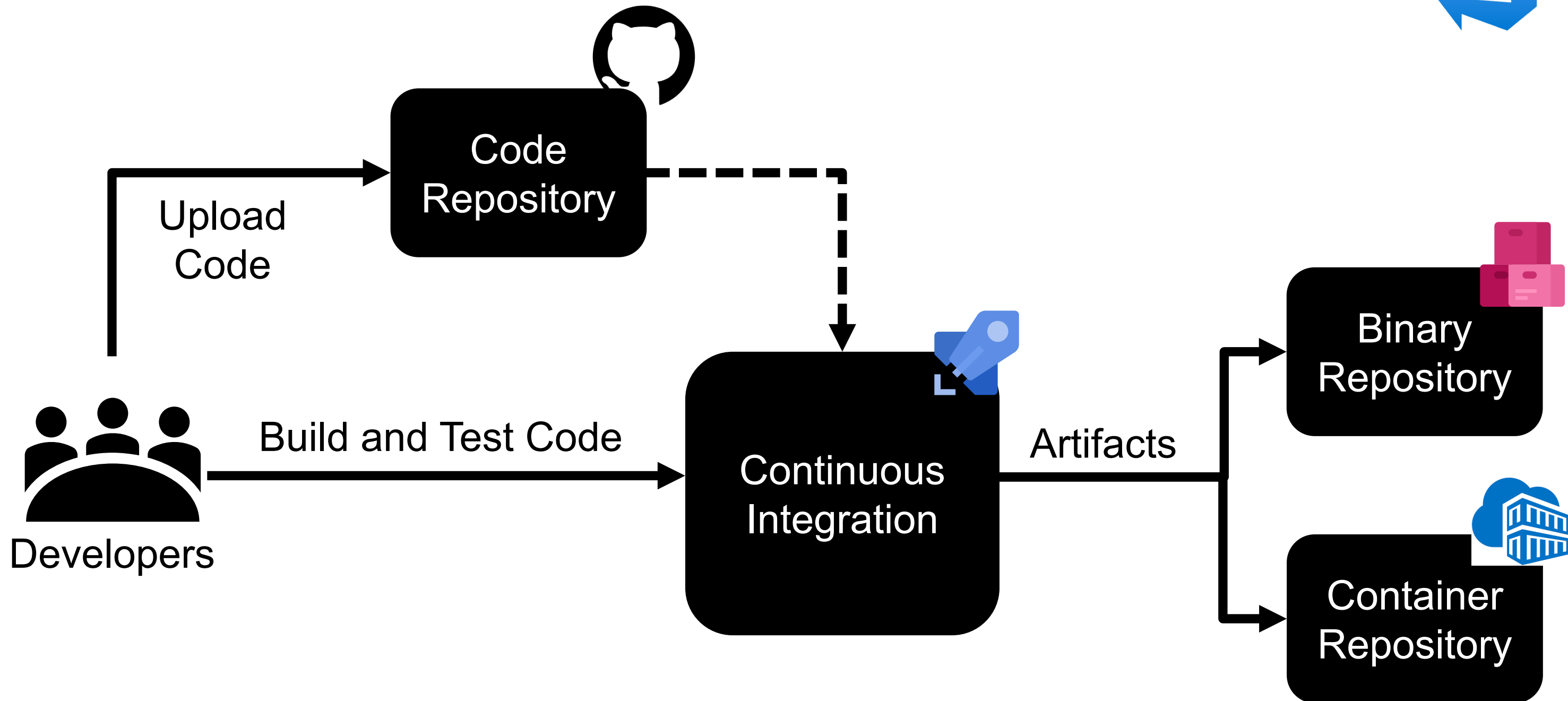
The Customer objective

- **Allow for Car OEM collaboration**
- **Scalability and Availability**
- **Performant; Developer feedback < 30 minutes**
- **Integrated and Secure**
- **Future proof**
- **Cost efficient**

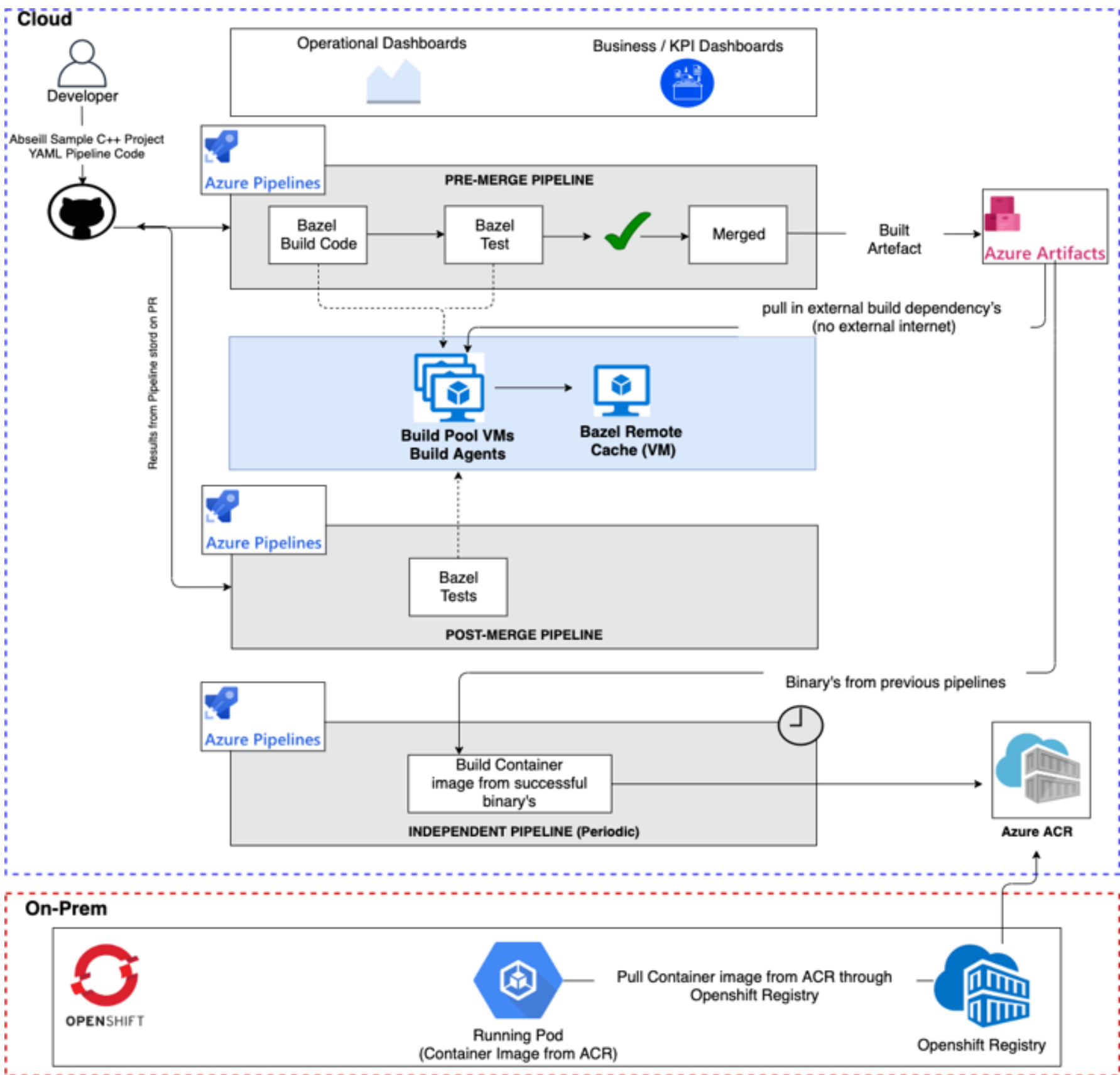
Since January 8, 2020

- **Defined criteria for successful POC at workshop on January 8**
- **Excellent team of DXC and MS developers**
 - DXC: ---- Confidential -----
 - Microsoft: ---- Confidential ----
- **Successfully implemented requested user stories in POC**
- **January 23: Demonstrate a successful POC in a couple of workflows**
- **PoC contents**
 - Infrastructure Overview
 - Workflow 1a: Commit code changes & pull request
 - Workflow 1b: Fixing failed pull requests
 - Workflow 2: Azure DevOps Portal (detailed look at logs, dashboards, boards)
 - Workflow 3a & b: Configuration as code (Pipelines and securities)
 - Workflow 4: Container on Premise
 - Optimizing Performance

Component Overview



Architecture



Addressed User Stories

✓ Also covers speculative merge during pre-merge

Status	As a <type of user>	I want to <perform some task/action>	So that I can <expect some result>
✓	Developer	login as user	have access
✓	Developer	define my pipeline definition as code in the same repo as the application	track configuration in repo and test config changes in PR (incl. tests)
✓	Developer	code review	approve PR before automatic merging to master
✓	Developer	trigger builds for any new PR raised on a code repo, before merge to master	builds are executed and result displayed on pipeline job
✓	Developer	trigger tests for any new PR raised on a code repo, after successful merge to master	tests are executed and result displayed on pipeline job
✓	Developer	trigger builds for any existing PR modified for a previously failing build on a code repo, before merge to master	builds and then tests are executed and result displayed on pipeline job
✓	Developer	for successful build completion and premerged tests	auto merge the PR into master and start running the postmerged tests
✓	Developer	access a live dashboard	every dev can see running jobs for all PR
✓	Developer	access a home page	every dev can see live logs for every running job

Addressed User Stories

Status	As a <type of user>	I want to <perform some task/action>	So that I can <expect some result>
✓	Developer	for every build or test triggered	display build/test graph on dashboard
✓	Developer	for every build or test triggered	provide access to logs on PR page
✓	Developer	for every build or test triggered	build or test dependency are referenced from Azure Artifacts
✓	Developer	for every completed successful build and test, for the specified bazel target(s)	binary artifact stored in Azure Artifacts
✓	Developer	perform a scheduled / user-triggered build	collect all binaries from previous builds (Azure Artifacts) and build a docker container and pushes it into an ACR
✓	Developer	access a status page	every dev can see results (including log files) of every previously run job and pipeline
✓	Developer	understand how we can host engineering tools	trace requirements back to specific PR

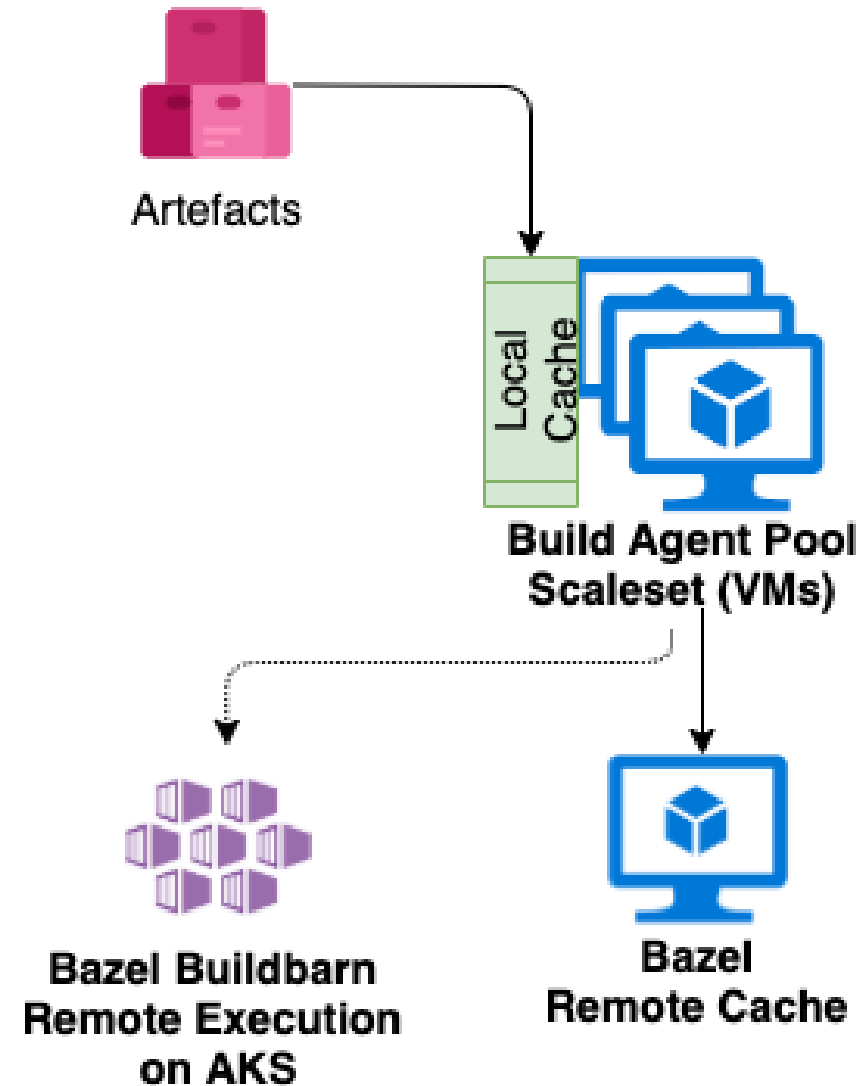
Addressed User Stories

Status	As a <type of user>	I want to <perform some task/action>	So that I can <expect some result>
✓	CI/CD Developer	code review of configuration changes	approve PR before automatic merging to master
✓	Developer	safely store credentials as part of the configuration code	verify password changes with a PR
✓	Developer	access container in ACR on premise	run the build container on an openshift platform
✓	Developer	compare build with and without populated remote cache	demonstrate performance improvement due to bazel remote cache integration
✓	Developer	reuse bazel actions and artifacts (git) from previous runs	download artifacts only once per build slave and day
✓	Developer	minimize start-up time (time between PR and start of bazel should be below 1 minute)	receive fast feedback
✓	Developer	understand from the PoC how we can extend environment to include build barn/remote execution	understand how to include it in production



Thank You

Timings for Configurations



Scenario	Local Cache Populated	Remote Cache Populated	Average Timing
BUILD	×	×	3m 30s
	×	✓	49s
	✓	×	22s
	✓	✓	4s
TEST	×	×	1m 10s
	×	✓	9s
	✓	×	5s
	✓	✓	<1s