



# New & Improved LLVM

## Premerge Testing: Status Update

# Brief Intro & History

# What is this Premerge Testing?

 **Review requested**  
Review has been requested on this pull request. It is not required to merge. [Learn more about requesting a pull request review.](#)

 **All checks have passed**  
2 skipped, 9 successful checks

 **9 successful checks**

-   buildkite/libcxx-ci — Build #89365 passed (2 seconds) 
-   buildkite/libcxx-ci/pipeline — Passed (2 seconds) 
-   Check code formatting / code\_formatter (pull\_request) Successful in 1m 
-   CI Checks / Build and Test Linux (pull\_request) Successful in 4m  
-   CI Checks / Build and Test Linux AArch64 (pull\_request) Successful in 7m 
-   CI Checks / Build and Test Windows (pull\_request) Successful in 2m  
-   Graphite / mergeability\_check 

 **No conflicts with base branch**   
Merging can be performed automatically.

**Squash and merge** You can also merge this with the command line. [View command line instructions.](#)

# What is this Premerge Testing?

A screenshot of a GitHub pull request interface. At the top left is a green 'Review requested' button with a checkmark icon. Below it, the status 'All checks have passed' is shown with a green circle icon. A summary indicates '2 skipped, 9 successful checks'. The list of successful checks includes:

- ✓ buildkite/libcxx-ci — Build #89365 passed (2 seconds)
- ✓ buildkite/libcxx-ci/pipeline — Passed (2 seconds)
- ✓ CI Checks / Build and Test Linux (pull\_request) Successful in 4m (Required)
- ✓ CI Checks / Build and Test Linux AArch64 (pull\_request) Successful in 7m (Required)
- ✓ CI Checks / Build and Test Windows (pull\_request) Successful in 2m (Required)
- ✓ Graphite / mergeability\_check

Below the checks, a section 'No conflicts with base branch' has a green checkmark icon. It says 'Merging can be performed automatically.' and features a 'Update branch' button. At the bottom, a green 'Squash and merge' button is available, along with a note: 'You can also merge this with the command line. [View command line instructions.](#)'

"Premerge Tests" for this talk.

# What is this Premerge Testing?

Not covered in this talk.

Review requested

Review has been requested on this pull request. It is not required to merge. [Learn more about requesting a pull request review.](#)

5 pending reviews >

All checks have passed

2 skipped, 9 successful checks

9 successful checks ▾

- ✓ buildkite/libcxx-ci — Build #89365 passed (2 seconds) ...
- ✓ buildkite/libcxx-ci/pipeline — Passed (2 seconds) ...
- ✓ Check code formatting / code\_formatter (pull\_request) Successful in 1m ...
- ✓ CI Checks / Build and Test Linux (pull\_request) Successful in 4m Required ...
- ✓ CI Checks / Build and Test Linux AArch64 (pull\_request) Successful in 7m Required ...
- ✓ CI Checks / Build and Test Windows (pull\_request) Successful in 2m ...
- ✓ Graphite / mergeability\_check ...

No conflicts with base branch

Merging can be performed automatically. [Update branch](#)

**Squash and merge** You can also merge this with the command line. [View command line instructions.](#)

# What is this Premerge Testing?

 **Review requested**  
Review has been requested on this pull request. It is not required to merge. [Learn more about requesting a pull request review.](#)

 **All checks have passed**  
2 skipped, 9 successful checks

 **9 successful checks**

-   buildkite/libcxx-ci — Build #89365 passed (2 seconds) 
-   buildkite/libcxx-ci/pipeline — Passed (2 seconds) 
-   Check code formatting / code\_formatter (pull\_request) Successful in 1m 
-   CI Checks / Build and Test Linux (pull\_request) Successful in 4m  
-   CI Checks / Build and Test Linux AArch64 (pull\_request) Successful in 7m 
-   CI Checks / Build and Test Windows (pull\_request) Successful in 2m  
-   Graphite / mergeability\_check 

 **No conflicts with base branch**   
Merging can be performed automatically.

**Squash and merge** You can also merge this with the command line. [View command line instructions.](#)

## Brief History: Old System

### OLD Premerge System (2019 - May 2025)

- Ran on BuildKite.
- Slow, flaky, unreliable.
- Not well maintained.

## Brief History: New System

### NEW Premerge System (May 2025+)

- Autoscaling
- Oncall rotation support
- Metrics & performance tracking
- Added the libc++ testing, early July

Overall goals: Speed, Reliability & Ease of Use.

- < 30 mins, for 80+% of tests
- No false positive error reports.
- Make failures easy to find & understand.

# New Features

# Finding & Understanding Test Failures (failure summary)

 Some checks were not successful

2 failing, 1 in progress, 3 skipped, 5 successful checks

2 failing checks ▾

-   CI Checks / Build and Test Linux (pull\_request) Failing after 4m Required ⋮
-   CI Checks / Build and Test Linux AArch64 (pull\_request) Failing after 10m ⋮

1 in progress check ▾

-   Test documentation build / Test documentation build (pull\_request) Started 33 minutes ago — This check has ... ⋮

3 skipped checks ▾

-   CI Checks / MacOS Premerge Checks (pull\_request) Skipped 33 minutes ago ⋮
-   Labelling new pull requests / automate-prs-labels (pull\_request\_target) Skipped 33 minutes ago ⋮
-   Labelling new pull requests / crocotor (pull\_request\_target) Skipped 33 minutes ago ⋮

1

1

# Finding & Understanding Test Failures (failure summary)

 Some checks were not successful

2 failing, 1 in progress, 3 skipped, 5 successful checks

2 failing checks ▾

-   CI Checks / Build and Test Linux (pull\_request) Failing after 4m Required 
-   CI Checks / Build and Test Linux AArch64 (pull\_request) Failing after 10m 

1 in progress check ▾

-   Test documentation build / Test documentation build (pull\_request) Started 33 minutes ago — This check has ... 

3 skipped checks ▾

-   CI Checks / MacOS Premerge Checks (pull\_request) Skipped 33 minutes ago 
-   Labelling new pull requests / automate-prs-labels (pull\_request\_target) Skipped 33 minutes ago 
-   Labelling new pull requests / crocotor (pull\_request\_target) Skipped 33 minutes ago 

# Finding & Understanding Test Failures (failure summary)

 Some checks were not successful

2 failing, 1 in progress, 3 skipped, 5 successful checks

2 failing checks ▾

-   CI Checks / Build and Test Linux (pull\_request) Failing after 4m Required ...
-   CI Checks / Build and Test Linux AArch64 (pull\_request) Failing after 10m ...

1 in progress check ▾

-   Test documentation build / Test documentation build (pull\_request) Started 33 minutes ago — This check has ... ...

3 skipped checks ▾

-   CI Checks / MacOS Premerge Checks (pull\_request) Skipped 33 minutes ago ...
-   Labelling new pull requests / automate-prs-labels (pull\_request\_target) Skipped 33 minutes ago ...
-   Labelling new pull requests / crocotor (pull\_request\_target) Skipped 33 minutes ago ...

1

1

1

1

1

1

1

# Finding & Understanding Test Failures (failure summary)

Some checks were not successful

2 failing, 1 in progress, 3 skipped, 5 successful checks

2 failing checks ▾

- ✗ CI Checks / Build and Test Linux (pull\_request) Failing after 4m Required ...
- ✗ CI Checks / Build and Test Linux AArch64 (pull\_request) Failing after 10m ...

1 in progress check ▾

- ⌚ Test documentation build / Test documentation build (pull\_request) Started 33 minutes ago — This check has ... ...

3 skipped checks ▾

- ⌚ CI Checks / MacOS Premerge Checks (pull\_request) Skipped 33 minutes ago ...
- ⌚ Labelling new pull requests / automate-prs-labels (pull\_request\_target) Skipped 33 minutes ago ...
- ⌚ Labelling now pull requests / crocotor (pull\_request\_target) Skipped 33 minutes ago ...

1

1

# Finding & Understanding Test Failures, Cont. (failure summary)

← Back to pull request #164291

**[LLDB] Temporarily break dil test #142681**

Summary

Jobs

- Build and Test Linux AArch64
- Build and Test Linux**
- Build and Test Windows
- MacOS Premerge Checks

Run details

Usage

Workflow file

Annotations  
1 error

**Build and Test Linux**  
failed 8 minutes ago in 4m 26s

> Set up job

> Checkout LLVM

> Build and Test

```
1 ► Run git config --global --add safe.directory '*'
41 Building projects: clang;lldb;llvm
42 Running project checks targets: check-lldb
43 Building runtimes: libcxx;libcxxabi;libunwind
44 Running runtimes checks targets:
45 Running runtimes checks requiring reconfiguring targets:
46 KUBERNETES_SERVICE_PORT_HTTPS=443
47 GITHUB_WORKSPACE=/home/gha/actions-runner/_work/llvm-project/llvm-project
48 KUBERNETES_SERVICE_PORT=443
49 HOSTNAME=llvm-premerge-linux-runners-ckfsp-runner-2pcpv
50 GITHUB_PATH=/home/gha/actions-runner/_work/_temp/_runner_file_commands/add_path_2aea957e-6bbd-450
51 GITHUB_ACTION=__run
52 GITHUB_RUN_NUMBER=142681
```

# Finding & Understanding Test Failures, Cont. (failure summary)

← Back to pull request #164291

## ✖ [LLDB] Temporarily break dil test #142681

[Summary](#) (highlighted with a red circle)

[Jobs](#)

- Build and Test Linux AArch64 (green checkmark)
- Build and Test Linux (red X)** (highlighted with a red rectangle)
- Build and Test Windows (green checkmark)
- MacOS Premerge Checks (grey circle)

[Run details](#)

[Usage](#)

[Workflow file](#)

[Annotations](#)  
1 error

**Build and Test Linux**  
failed 8 minutes ago in 4m 26s

- Set up job (green checkmark)
- Checkout LLVM (green checkmark)
- Build and Test (red X)** (highlighted with a red rectangle)

```
1 ► Run git config --global --add safe.directory '*'
41 Building projects: clang;lldb;llvm
42 Running project checks targets: check-lldb
43 Building runtimes: libcxx;libcxxabi;libunwind
44 Running runtimes checks targets:
45 Running runtimes checks requiring reconfiguring targets:
46 KUBERNETES_SERVICE_PORT_HTTPS=443
47 GITHUB_WORKSPACE=/home/gha/actions-runner/_work/llvm-project/llvm-project
48 KUBERNETES_SERVICE_PORT=443
49 HOSTNAME=llvm-premerge-linux-runners-ckfsp-runner-2pcpv
50 GITHUB_PATH=/home/gha/actions-runner/_work/_temp/_runner_file_commands/add_path_2aea957e-6bbd-450
51 GITHUB_ACTION=__run
52 GITHUB_RUN_NUMBER=142681
```

# Finding & Understanding Test Failures, Cont. (failure summary)

## Jobs

- Build and Test Linux AArch64
- Build and Test Linux
- Build and Test Windows
- MacOS Premerge Checks

## Run details

Usage

Workflow file

## Build and Test Linux summary

...

## Linux x64 Test Results

- 33068 tests passed
- 478 tests skipped
- 3 tests failed

## Failed Tests

(click on a test name to see its output)

[lldb-api](#)

- ▶ lldb-api.commands/frame/var-dil/basics/ArraySubscript/TestFrameVarDILArraySubscript.py
- ▶ lldb-api.commands/frame/var-dil/basics/LocalVars/TestFrameVarDILLocalVars.py
- ▶ lldb-api.commands/frame/var-dil/basics/MemberOfInheritance/TestFrameVarDILMemberOfInheritance.py

If these failures are unrelated to your changes (for example tests are broken or flaky at HEAD), please open an issue at <https://github.com/llvm/llvm-project/issues> and add the `infrastructure` label.

Job summary generated at run-time

## Annotations

# Finding & Understanding Test Failures, Cont. (failure summary)

lldb-api

► lldb-api.commands/frame/var-dil/basics/ArraySubscript/TestFrameVarDILArraySubscript.py

► lldb-api.commands/frame/var-dil/basics/LocalVars/TestFrameVarDILLocalVars.py

▼ lldb-api.commands/frame/var-dil/basics/MemberOfInheritance/TestFrameVarDILMemberOfInheritance.py

Script:

```
-- /usr/bin/python3 /home/gha/actions-runner/_work/llvm-project/llvm-project/lldb/test/API/dotest.py -u CXXFLAGS -u CFLAGS --env LLVM_LIBS
```

--

Exit Code: 1

Command Output (stdout):

```
-- lldb version 22.0.0git (https://github.com/llvm/llvm-project revision 6164c521e1e9f82741911db872b2a44f1487966c)
  clang revision 6164c521e1e9f82741911db872b2a44f1487966c
  llvm revision 6164c521e1e9f82741911db872b2a44f1487966c
Skipping the following test categories: ['msvcstl', 'dsym', 'gmodules', 'debugserver', 'objc']
```

--

Command Output (stderr):

```
-- FAIL: LLDB (/home/gha/actions-runner/_work/llvm-project/llvm-project/build/bin/clang-x86_64) :: test_frame_var (TestFrameVarDILMemberOfInheritance.TestFrameVarDILMemberOfInheritance.test_frame_var)
=====
FAIL: test_frame_var (TestFrameVarDILMemberOfInheritance.TestFrameVarDILMemberOfInheritance.test_frame_var)
```

Traceback (most recent call last):

```
  File "/home/gha/actions-runner/_work/llvm-project/llvm-project/lldb/test/API/commands/frame/var-dil/basics/MemberOfInheritance/TestFrameVarDILMemberOfInheritance.py", line 10, in <module>
    self.expect_var_path("parent->z", value="5")
  File "/home/gha/actions-runner/_work/llvm-project/llvm-project/lldb/packages/Python/lldbsuite/test/lldbtest.py", line 2599, in expect_var_path
    value_check.check_value(self, eval_result, str(eval_result))
  File "/home/gha/actions-runner/_work/llvm-project/llvm-project/lldb/packages/Python/lldbsuite/test/lldbtest.py", line 313, in check_value
    test_base.assertEqual(self.expect_value, val.GetValue(), this_error_msg)
AssertionError: '5' != '3'
- 5
+ 3
: (int) z = 3
Checking SBValue: (int) z = 3
Config=x86_64-/home/gha/actions-runner/_work/llvm-project/llvm-project/build/bin/clang
```

Ran 1 test in 0.457s

# Don't have to wait for tests to finish!

The screenshot shows a GitHub pull request merge interface with the following sections:

- Changes reviewed**: 1 approving review by reviewers with write access.
  - ✓ 1 approval >
  - ✗ 1 pending review >
- Some checks haven't completed yet**: 3 in progress, 2 skipped, 5 successful checks
  - 3 in progress checks ▾
    - 🟡 CI Checks / Build and Test Linux (pull\_request) Started 15 minutes ago — This check has started... **Required** ⋮
    - 🟡 CI Checks / Build and Test Linux AArch64 (pull\_request) Started 16 minutes ago — This check has started... ⋮
    - 🟡 CI Checks / Build and Test Windows (pull\_request) Started 15 minutes ago — This check has started... **Required** ⋮
  - 2 skipped checks ▾
    - 🚫 CI Checks / MacOS Premerge Checks (pull\_request) Skipped 16 minutes ago ⋮
    - 🚫 Labelling new pull requests / greeter (pull\_request\_target) Skipped 16 minutes ago ⋮
  - 5 successful checks ▾
    - 🟢 buildkit/libuv ci Build #600286 passed (2 seconds) ⋮
- This branch is out-of-date with the base branch**: Merge the latest changes from main into this branch. This merge commit will be associated with boomaniden154.
  - Update branch**
  - Merge without waiting for requirements to be met (bypass rules)
- Enable auto-merge (squash)**: You can also merge this with the command line. [View command line instructions.](#)

# Don't have to wait for tests to finish!

Option 1:  
Merge  
before  
they're  
done.

The screenshot shows a GitHub pull request merge interface. At the top, it says "Changes reviewed" with "1 approving review by reviewers with write access". Below that, there are sections for "Some checks haven't completed yet" (3 in progress, 2 skipped, 5 successful checks), "3 in progress checks" (CI Checks / Build and Test Linux (pull\_request), CI Checks / Build and Test Linux AArch64 (pull\_request), CI Checks / Build and Test Windows (pull\_request)), and "2 skipped checks" (CI Checks / MacOS Premerge Checks (pull\_request), Labelling new pull requests / greeter (pull\_request\_target)). At the bottom, it says "This branch is out-of-date with the base branch" and provides an "Update branch" button. There is also a checkbox for "Merge without waiting for requirements to be met (bypass rules)".

Changes reviewed  
1 approving review by reviewers with write access.

✓ 1 approval >  
✗ 1 pending review >

Some checks haven't completed yet  
3 in progress, 2 skipped, 5 successful checks

3 in progress checks ▾  
CI Checks / Build and Test Linux (pull\_request) Started 15 minutes ago — This check has started... Required ...  
CI Checks / Build and Test Linux AArch64 (pull\_request) Started 16 minutes ago — This check has started... ...  
CI Checks / Build and Test Windows (pull\_request) Started 15 minutes ago — This check has started... Required ...

2 skipped checks ▾  
CI Checks / MacOS Premerge Checks (pull\_request) Skipped 16 minutes ago ...  
Labelling new pull requests / greeter (pull\_request\_target) Skipped 16 minutes ago ...

5 successful checks ▾  
buildkit/libuv ci Build #00028E passed (2 seconds) ...

This branch is out-of-date with the base branch  
Merge the latest changes from main into this branch. This merge commit will be associated with boomaniden154.  
Update branch

Merge without waiting for requirements to be met (bypass rules)

Enable auto-merge (squash) You can also merge this with the command line. [View command line instructions](#).

# Don't have to wait for tests to finish!

Option 1:  
Merge  
before  
they're  
done.

The screenshot shows a GitHub pull request interface with the following sections:

- Changes reviewed**: 1 approving review by reviewers with write access.
  - ✓ 1 approval >
  - ✗ 1 pending review >
- Some checks haven't completed yet**: 3 in progress, 2 skipped, 5 successful checks
  - 3 in progress checks ▾
    - 🟡 CI Checks / Build and Test Linux (pull\_request) Started 15 minutes ago — This check has started... **Required** ⋮
    - 🟡 CI Checks / Build and Test Linux AArch64 (pull\_request) Started 16 minutes ago — This check has started... ⋮
    - 🟡 CI Checks / Build and Test Windows (pull\_request) Started 15 minutes ago — This check has started... **Required** ⋮
  - 2 skipped checks ▾
    - ⓧ CI Checks / MacOS Premerge Checks (pull\_request) Skipped 16 minutes ago ⋮
    - ⓧ Labelling new pull requests / greeter (pull\_request\_target) Skipped 16 minutes ago ⋮
  - 5 successful checks ▾
    - Buildkitlibev ci Build #600286 passed (2 seconds)
- This branch is out-of-date with the base branch**: Merge the latest changes from main into this branch. This merge commit will be associated with boamaniden154.
  - Merge without waiting for requirements to be met (bypass rules)
- Enable auto-merge (squash)**: You can also merge this with the command line. [View command line instructions](#).

Option 2:  
Auto-merge  
when they  
finish.

# Performance Improvements

# Metrics & Performance Monitoring: Linux & Windows Current Performance



<https://llvm.grafana.net/public-dashboards/21c6e0a7cdd14651a90e118df46be4cc>

# Metrics & Performance Monitoring: Libc++ Current Performance



<https://llvm.grafana.net/public-dashboards/0bd453e8b3034733a1b0ff8c7728086d>

# Metrics & Performance Monitoring: Linux & Windows Performance Over Time



<https://llvm.grafana.net/public-dashboards/106344f8f6f649c2972a57f201db3a65>

## Performance Improvement: Caching objects.

- Updated testing to cache built objects.
- Re-uses cached objects whenever possible.
- Uses sccache in GCS (Google Cloud Storage)
- => 30-35% reduction in overall testing times!

# Performance Improvement (Linux): Migrating tests to lit internal shell

- **What:** Lit internal shell:
  - Limited functionality shell (compared to bash)
  - Avoids spawning test shell in another process
- **Why?**
  - => 10-15% performance improvement on Linux
  - (Windows already using it!)
- **When?**
  - Most tests already using it, including clang and LLVM.
  - In progress: compiler-rt tests.

## Performance Improvement (Windows): Switch to using Clang.

- Coming soon: Work-in-progress.
- Expect 40-50% reduction in compilation time...
  - (Preliminary results!)

# Flakiness Reduction

# Flakiness Reduction: Postsubmit Testing

- Need to know what's failing at head
  - => Don't want to report those as PR failures.
- Created a "postsubmit" test buildbot.
  - Runs same tests as premerge testing..
  - Same configuration as premerge, but builds/tests all projects every run.
  - Results get uploaded for comparison.
  - Runs continuously.

Builder Name	Builds	Tags	Workers
premerge-monolithic-linux	<a href="#">8538</a> <a href="#">8537</a> <a href="#">8536</a> <a href="#">8535</a> <a href="#">8534</a> <a href="#">8533</a> <a href="#">8532</a> <a href="#">8531</a> <a href="#">8530</a> <a href="#">8529</a> <a href="#">8528</a> <a href="#">8527</a> <a href="#">8526</a> <a href="#">8525</a> <a href="#">8524</a>		<a href="#">182</a> <a href="#">183</a> <a href="#">184</a> <a href="#">189</a> <a href="#">190</a> <a href="#">191</a>
premerge-monolithic-windows	<a href="#">7061</a> <a href="#">7060</a> <a href="#">7059</a> <a href="#">7058</a> <a href="#">7057</a> <a href="#">7056</a> <a href="#">7055</a> <a href="#">7054</a> <a href="#">7053</a> <a href="#">7052</a> <a href="#">7051</a> <a href="#">7050</a> <a href="#">7049</a> <a href="#">7048</a> <a href="#">7047</a>		<a href="#">180</a> <a href="#">181</a> <a href="#">185</a> <a href="#">186</a> <a href="#">187</a> <a href="#">188</a>

## Flakiness Reduction: Premerge Advisor

- Checks failing premerge tests.
- Eliminates tests known to fail at head
  - (from postsubmit buildbot data)
- Eliminates known flaky tests.
- If any failures are left, reports tests as failing.
- Otherwise reports tests as passing.

GOAL: Eliminate false positive failure reports!

## Flakiness Reduction: Future Plans

- Expose flaky test information from the premerge advisor.
  - Gives accurate flake data for the common platforms.
  - Makes it easy to report issues upstream to test authors.

Ideally: eliminate all test flakiness on the most common platforms.

Thank you!