Saucelabs testing with Bazel

Local testing

1. Set up your `SAUCE_USERNAME`, `SAUCE_ACCESS_KEY` & `SAUCE_TUNNEL_IDENTIFIER` environment variables. These are required. You can find the values for `SAUCE_USERNAME` and `SAUCE_ACCESS_KEY` in `/.circleci/env.sh`. `SAUCE_TUNNEL_IDENTIFIER` can be set to any unique value.

If you are having trouble running Saucelabs tests locally you can contact Joey Perrott or Greg Magolan for support.

1. On OSX and Windows, you will also need to set `SAUCE_CONNECT` to the path of your `sc` binary (Sauce Connect Proxy). You will have to download Sauce Connect Proxy if you don't already have it downloaded. It's available on the SauceLabs website here. Unzip it and point the SAUCE_CONNECT env variable to the `sc` binary.

```
export SAUCE_CONNECT=/{path_to_sc}/bin/sc
```

Note: it will not work to use the Sauce Connect that's already in node_modules unless you are using Linux. Download the one above for other platforms.

3. Once you have your environment variables set up, run the setup task in the root of the repo:

```
yarn bazel run //tools/saucelabs:sauce_service_setup
```

4. You can run a particular test target through SauceLabs by prefixing the target name with "saucelabs_" and adding the `--config=saucelabs` option. For example, `packages/core/test:test_web` becomes `packages/core/test:saucelabs_test_web`.

```
yarn bazel test //packages/core/test:saucelabs_test_web --config=saucelabs
```

5. Sauce service log may be tailed or dumped with the following targets:

```
yarn bazel run //tools/saucelabs:sauce_service_tail
yarn bazel run //tools/saucelabs:sauce_service_log
```

Additional test features

To see the test output while the tests are running (as these are long tests), add the `--test_output=streamed` option. Note, this option will also prevent bazel from using the test cache and will force the test to run.

`bazel query` is required gather up all karma saucelabs test labels so they can be run in one command as they are tagged `manual`.

Running all karma tests in Saucelabs:

```
yarn bazel run //tools/saucelabs:sauce_service_setup
TESTS=$(./node_modules/.bin/bazelisk query --output label '(kind(karma_web_test,
...) intersect attr("tags", "saucelabs", ...)) except attr("tags", "fixme-
saucelabs", ...)')
yarn bazel test --config=saucelabs ${TESTS}
```

Under the hood

The `//tools/saucelabs:sauce_service_setup` target does not start the Sauce Connect proxy but it does start the process that then listens for the start signal from the service manager script. This signal is sent by the karma wrapper script `//tools/saucelabs:karma-saucelabs` which calls `./tools/saucelabs/sauce-service.sh start`. This is necessary as the Sauce Connect Proxy process must be started outside of `bazel test` as Bazel will automatically kill any processes spawned during a test when that tests completes, which would prevent the tunnel from being shared by multiple tests.

The karma_web_test rule is to test with saucelabs with a modified `karma` attribute set to `//tools/saucelabs:karma-saucelabs`. This runs the `/tools/saucelabs/karma-saucelabs.js` wrapper script which configures the saucelabs environment and starts Sauce Connect before running karma.

For example,

```
karma_web_test(
    name = "saucelabs_core_acceptance_tests",
    timeout = "long",
    karma = "//tools/saucelabs:karma-saucelabs",
    tags = [
        "exclusive",
        "manual",
        "no-remote-exec",
        "saucelabs",
    ],
    deps = [
        "//packages/core/test/acceptance:acceptance_lib",
    ],
)
```

These saucelabs targets must have a few important tags:

- `no-remote-exec` as they cannot be executed remotely since they require a local Sauce Connect process
- `manual` so they are not automatically tested with `//...`
- `exclusive` as they must be run serially in order to not over-provision Saucelabs browsers
- `saucelabs` so that they can be easily gathered up for testing in a `bazel query`

Debugging

Q: How do I get the tests to run on IE? I only see Chromium.

If you see something like this at the end of your test output, it means you're not actually running SauceLabs:

```
INFO: Build completed successfully, 43 total actions
/packages/core/test:test_web_chromium
```

This is a common error caused by forgetting to prefix your test target with "saucelabs_". For example, `/packages/core/test:test_web` becomes `/packages/core/test:saucelabs_test_web`.

Q: How can I tell that the SauceLabs connection was successfully made?

There is a dashboard at saucelabs.com where you can see active tunnel connections (Angular has an account). As soon as you actually run the test target (not after the setup task), you should see an active tunnel connection under the SAUCE_TUNNEL_IDENTIFICATION_KEY you entered. If a tunnel connection is not there, you are not actually connecting with SauceLabs.

Note: It may *look* like the tests are running because of the Bazel output. The progress Bazel is showing does not mean that SauceLabs is connected. If the tests are actually running, you should see the "..." test report for passing tests.