Meteor distributes fully pre-build packages called Isopacks. This helps ensure similar behavior across platforms and means that casual Meteor users do not need to install Xcode in order to run their apps. This means that packages must be built for all platforms that are supported by Meteor (os.osx.x86_64, os.linux.x86_64 and os.linux.x86_32). In the future, this will be done automatically by Meteor – but for now, we ask that you publish the builds manually.

In order to help with that, Meteor provides access to pre-configured build machines. To access a build machine, log in with your meteor developer account and run:

```
# OS X
meteor admin get-machine os.osx.x86_64

# Linux on 64-bit Intel
meteor admin get-machine os.linux.x86_64

# Linux on 32-bit Intel
meteor admin get-machine os.linux.x86_32

# Windows on 32-bit Intel
meteor admin get-machine os.windows.x86_32
```

This will open a secure shell to build machines on all three platforms. From there, you can use `meteor publish-for-arch` to publish the package; see `meteor help publish-for-arch` for more details.

For more on Meteor Build, see https://www.meteor.com/services/build.

FAQ

Why does Meteor think that my package is not platform-agnostic?

Your package probably contains binary code. Most likely, this is because:

- Your package includes a binary NPM module. If your package is in active development, one pattern is to move the NPM dependencies to a different package and version them separately. (To use the core packages as an example, we version `npm-bcrypt` separately from the `accounts` packages, so when we change accounts code, the package we only have to publish a new version of the js-only package)

- Your package includes a build plugin that uses binary code in some way. A build plugin extends the build process for apps and packages that use this package. Because of this, it is compiled together with its dependencies. So, having a build plugin depend on `npm-bcrypt` will pull in the binary

contents of `npm-bcrypt`, and cause the build plugin (and the package containing the build plugin) to contain binary code.

Why is it important to use pre-configured build machines? Why shouldn't I just publish from my laptop?

Meteor, and, by extension, your package, has many users all around the world. Many of them are running older version of operating systems, using older compilers, etc. These are usually backwards, but not forwards compatible, so a package compiled on a new machine, might throw errors on old machines.

As an example, here is a github issue that was caused by compiling a package on a machine with the newer version of glibc (https://github.com/meteor/meteor/issues/2554#issuecomment-55264224) We have set up build machines to both avoid this sort of thing, and be able to take responsibility for them if they happen again.

Can you give me more details on get-machine? What are the limits?

Here is how `meteor admin get-machine` currently works on the client: * You ask for a machine. * The server assigns you a machine for a specified (5 minutes default) length of time. * The server passes you login information. * If, within that length of time, you run the command again, you will be assigned the same machine.

Here is how it works on the server: * There is a set amount of machines/VMs that have mostly been spun up and ready to go. * When a user asks for one, one is assigned to them for a length of time. * Once you are assigned a machine, no other user may have it. When the server decides that you are done, so is the machine. * Sometimes, for technical/economics reasons, it doesn't make much sense to kill the machine immediately. So, you might have requested a machine for five minutes, but get to keep it for up to an hour (and keep getting the same machine). That's cool, but you should not rely on that behavior.

Here are some time limits: * You cannot reserve a machine for more than 15 minutes at a time. * You cannot have more than 60 minutes of machine-usage per week.

We are still experimenting with these numbers. If you are a package author and you need more time, please let us know what we can do.