App Testing at Scale: How Bitnami Tests Thousands of Releases Per Month

Juan Jose Martos Castro

Bitnami, Now part of VMware

KubeCon + CloudNativeCon North America

Bitnami is a Catalog of Free Open Source Software

10+ years building and maintaining software packages

Over 180 applications, components, frameworks, templates, and more, including…

Any Environment

Local

Cloud

Data Center

Any Format

Virtual Machines

Containers

Deployment Templates

Any Platform

vmware

aws

ORACLE

Azure

Google

IBM Cloud

Testing and publishing the catalog is hard…

500

Components we track

2500

Tests we use

12000

Containers we release every month

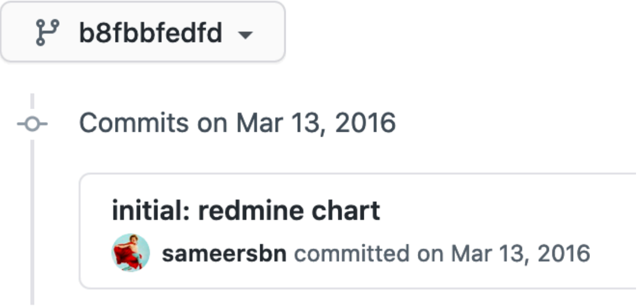
75

Charts we maintain

…and requires experience

Community offering: More than 10 years of backlog publishing the solutions in the major cloud providers

And more than 4 years generating Charts for the users


b8fbbfedfd

Commits on Mar 13, 2016

initial: redmine chart

sameersbn committed on Mar 13, 2016

Enterprise offering (VMware Tanzu)

- Open source and commercial applications

- Customized to their specific requirements

- Multiple platforms

Our pillars

Up-to-date

Always using the latest versions

Security

Taking the best security practices in mind

Unified configuration

To simplify the development on top of our solutions

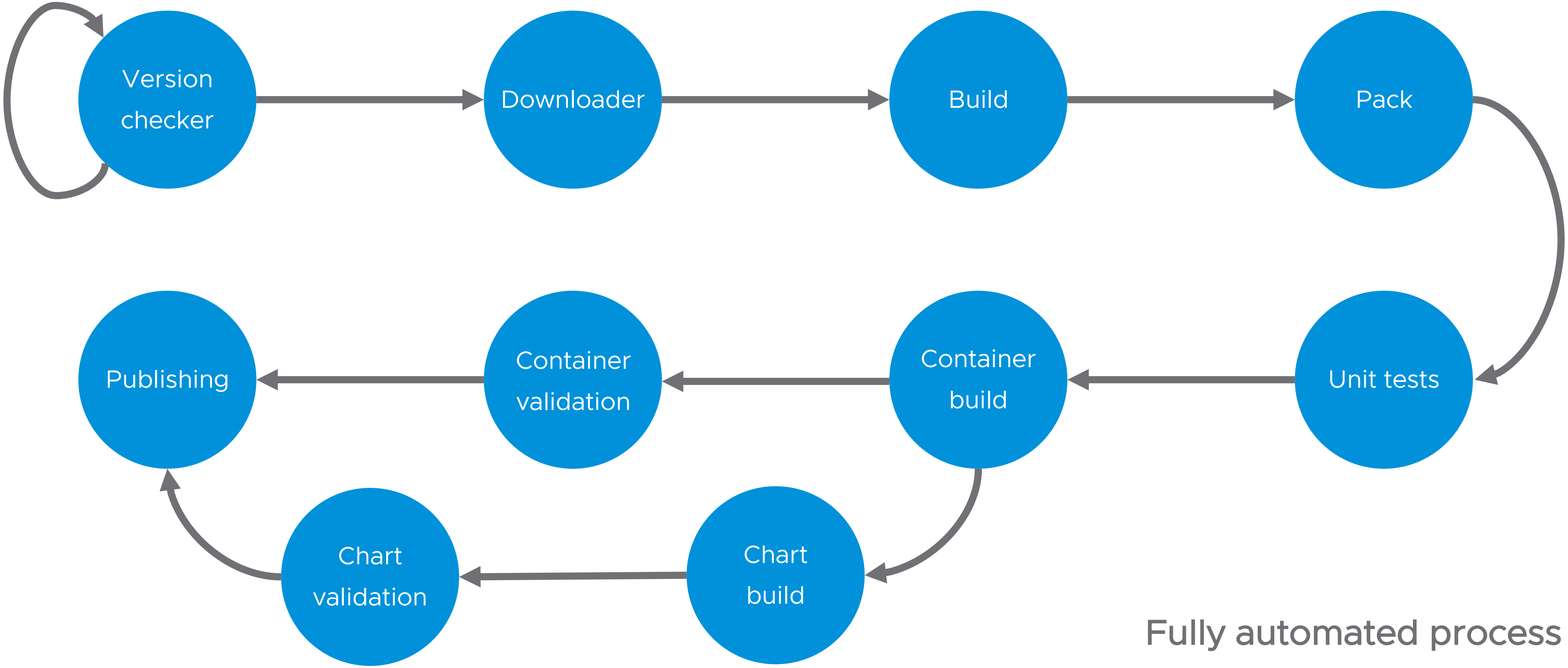
Useful documentation

To help our users to perform common actions

High quality support

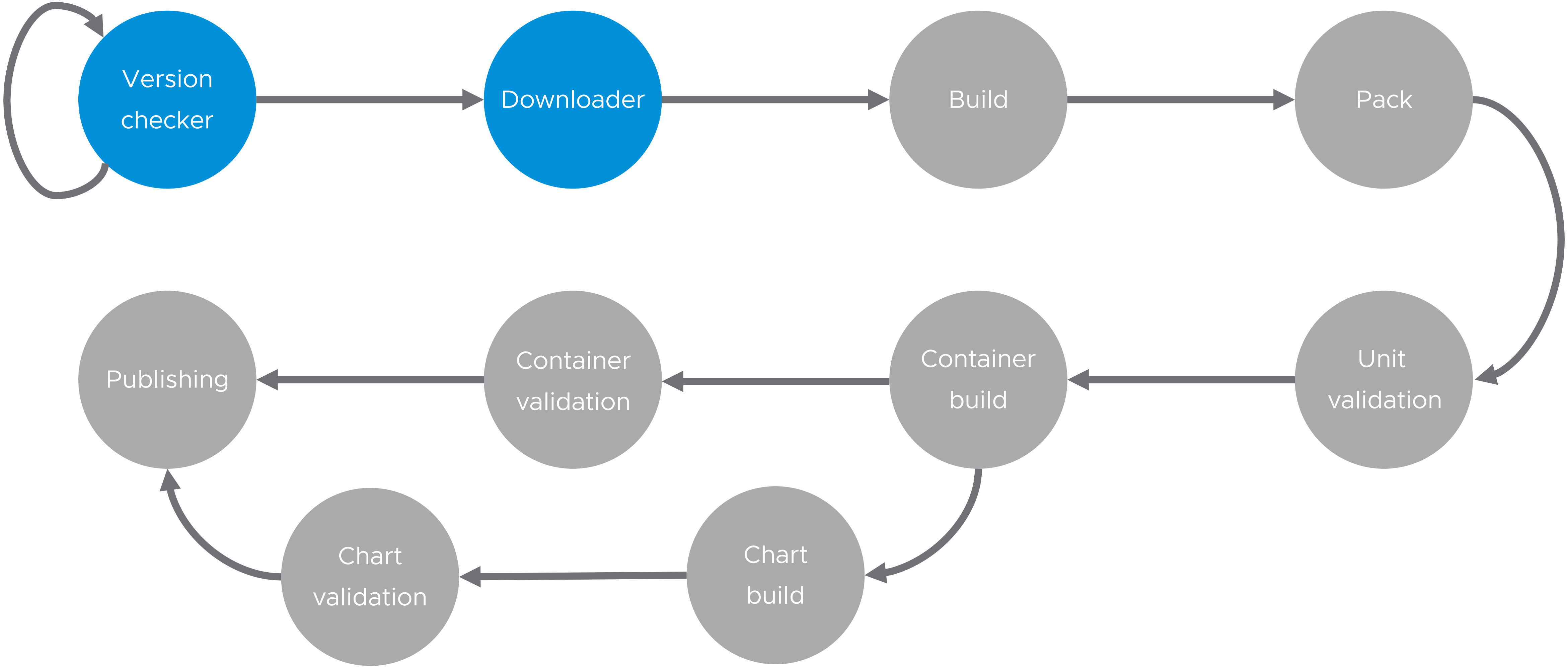
To answer any questions the users have

Automation is key



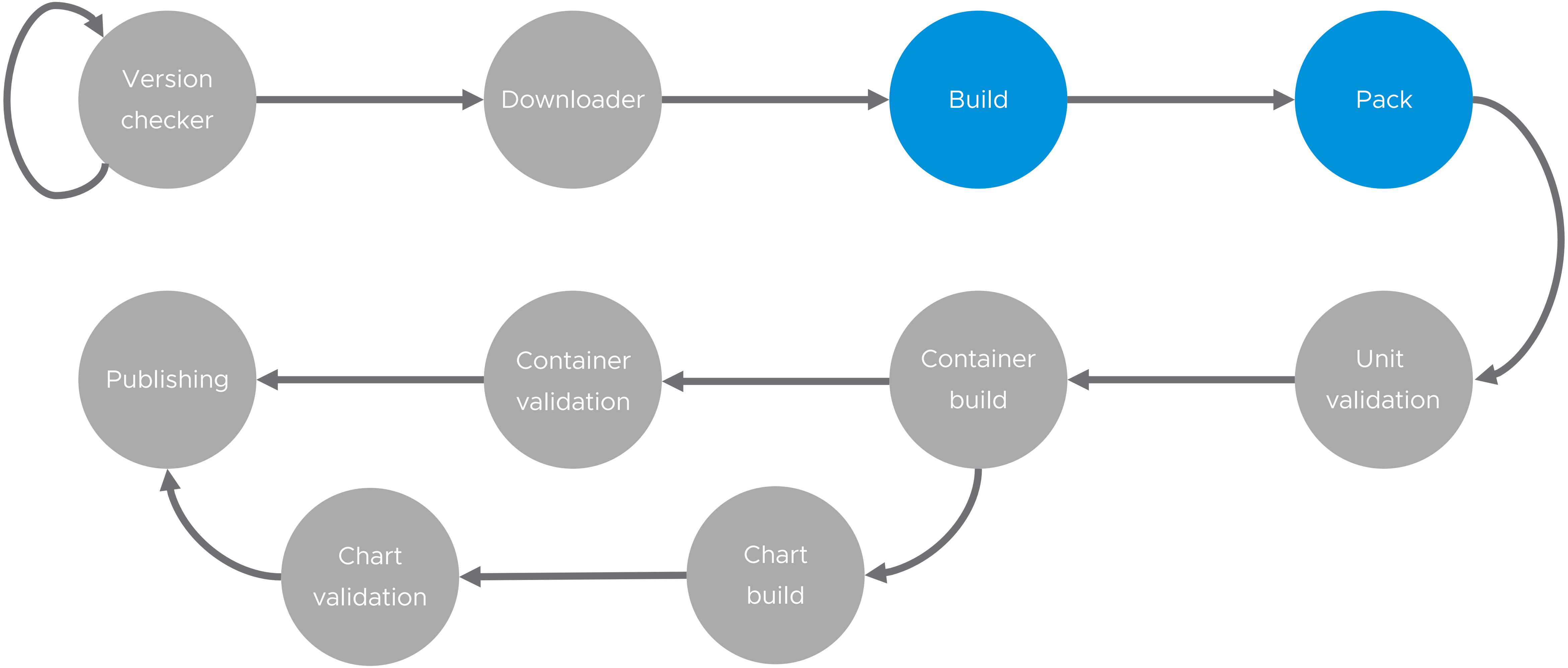
Fully automated process

Automation is key



Automation is key

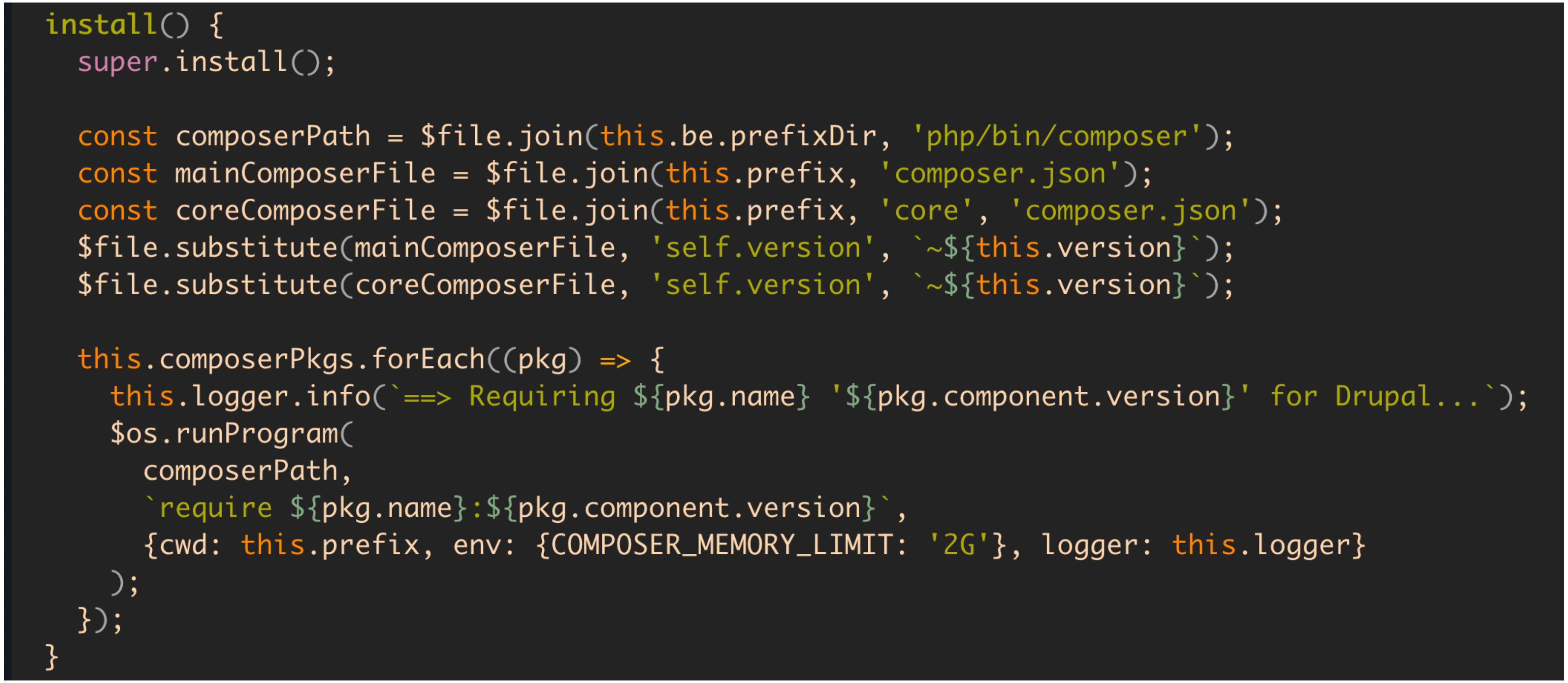
Build process



Automation is key

Build process

Prepare the package for a later usage



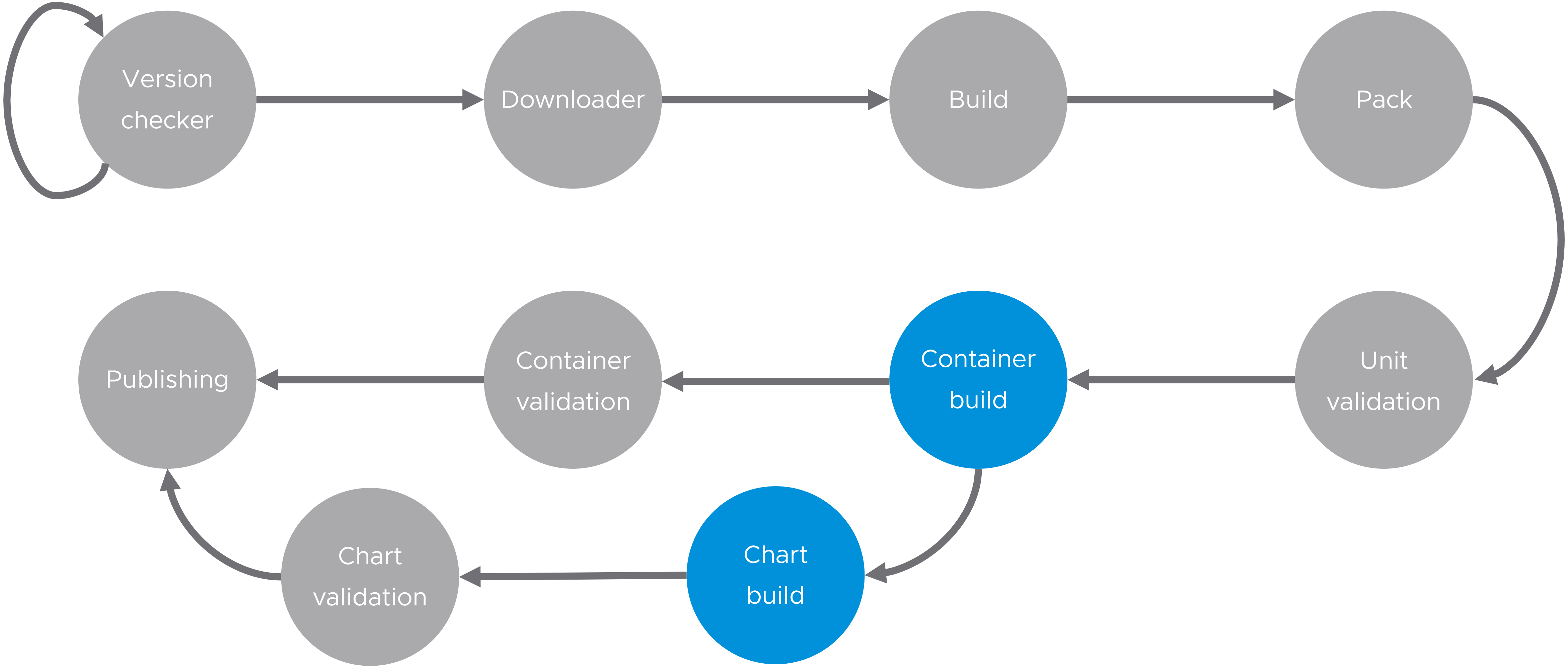
```
install() {
  super.install();

  const composerPath = $file.join(this.be.prefixDir, 'php/bin/composer');
  const mainComposerFile = $file.join(this.prefix, 'composer.json');
  const coreComposerFile = $file.join(this.prefix, 'core', 'composer.json');
  $file.substitute(mainComposerFile, 'self.version', `~${this.version}`);
  $file.substitute(coreComposerFile, 'self.version', `~${this.version}`);

  this.composerPkgs.forEach((pkg) => {
    this.logger.info(`==> Requiring ${pkg.name} '${pkg.component.version}' for Drupal...`);
    $os.runProgram(
      composerPath,
      `require ${pkg.name}:${pkg.component.version}`,
      {cwd: this.prefix, env: {COMPOSER_MEMORY_LIMIT: '2G'}, logger: this.logger}
    );
  });
}
```

Automation is key

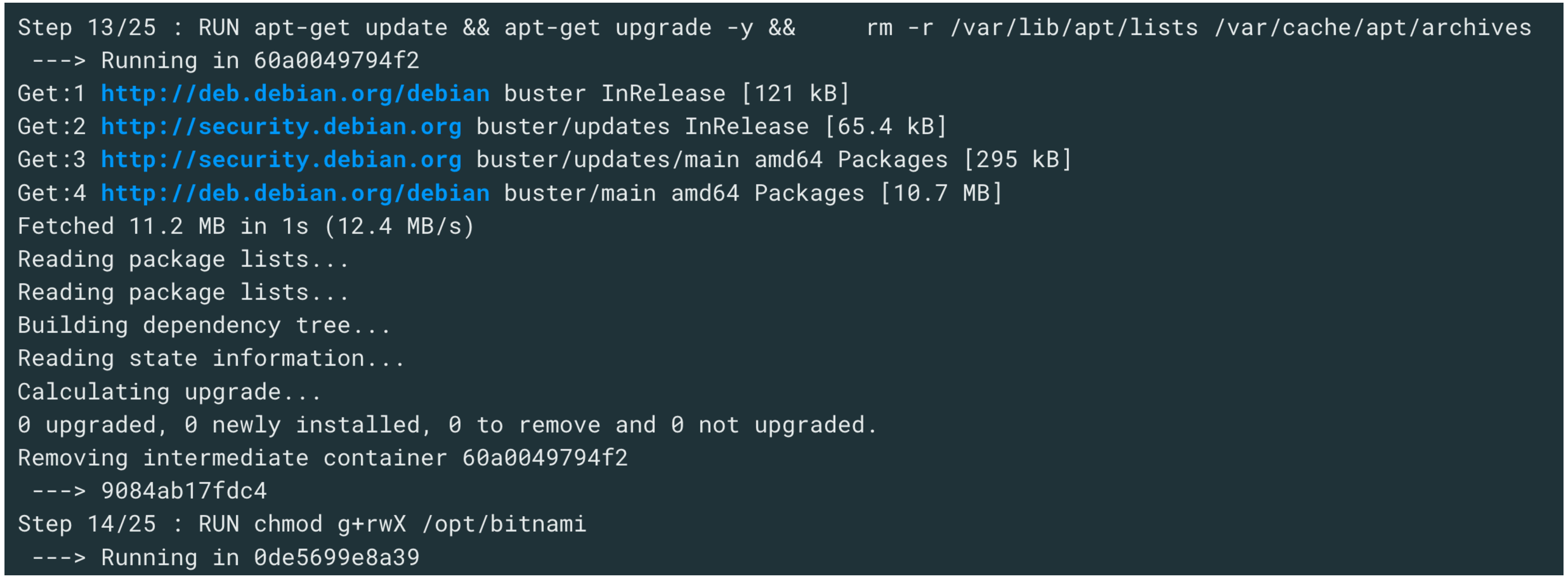
Build process



Automation is key

Build process

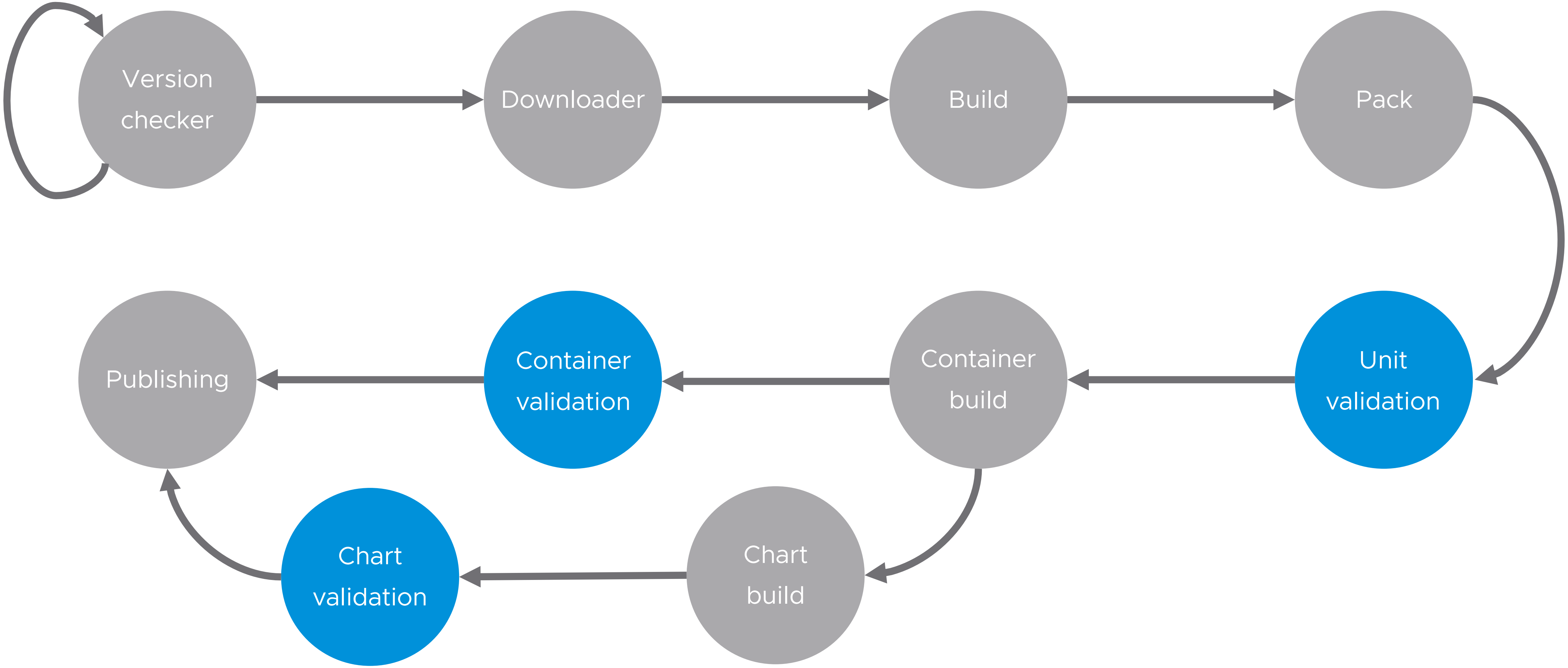
- Updates Dockerfile to use the latest versions of Bitnami packages as well as updating the system's packages
- Updates the Chart's files to use the new testing containers



```
Step 13/25 : RUN apt-get update && apt-get upgrade -y &&     rm -r /var/lib/apt/lists /var/cache/apt/archives
 ---> Running in 60a0049794f2
Get:1 http://deb.debian.org/debian buster InRelease [121 kB]
Get:2 http://security.debian.org buster/updates InRelease [65.4 kB]
Get:3 http://security.debian.org buster/updates/main amd64 Packages [295 kB]
Get:4 http://deb.debian.org/debian buster/main amd64 Packages [10.7 MB]
Fetched 11.2 MB in 1s (12.4 MB/s)
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Removing intermediate container 60a0049794f2
 ---> 9084ab17fdc4
Step 14/25 : RUN chmod g+rwX /opt/bitnami
 ---> Running in 0de5699e8a39
```

Automation is key

Test process



Automation is key

Test process

The Bitnami Validation System is a set of tools, scripts and files

+

that defines the actions to perform

Common actions:

- Confirm that there are no CVEs affecting the solution
- Deploy the solution and confirm that it is deployed correctly
- Perform the upgrade process
- Log in to the application with the random password we generated
- Create a new site/job/report/etc. or install a plugin/theme
- Run one command and ensure it shows the correct output
- Check that the permissions of this file are correct
- etc.

when validating the different applications

in all the supported platforms

Charts

- TKG, GKE, AKS, IKS

Containers

- Dockerhub, AWS

Virtual Machines

SingleVMs

- VMware, AWS, Google, Azure, IBM

MultiTier

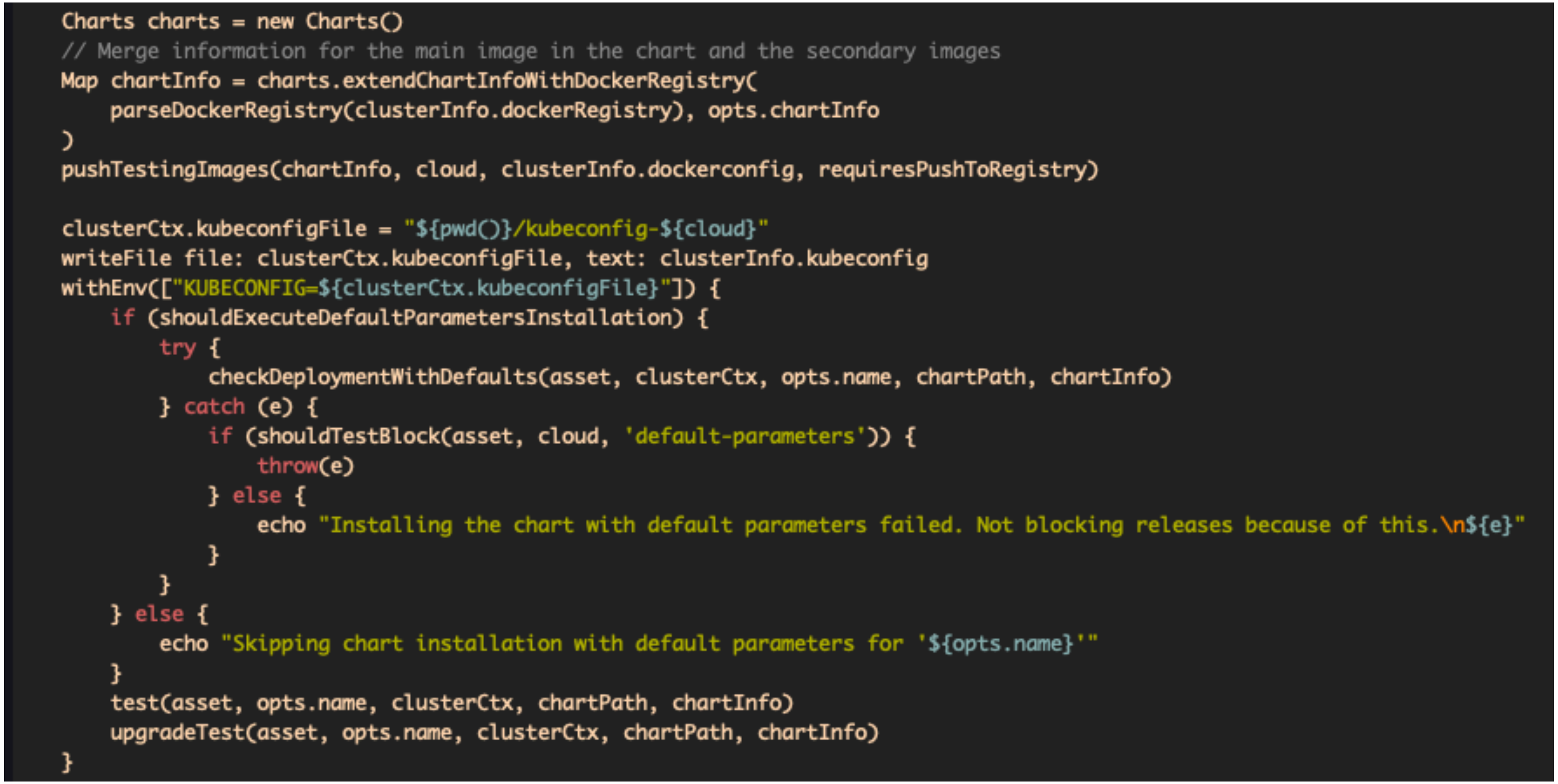
- AWS, Google, Azure, Vmware

Native Installers

- Windows, Linux and OS X

Automation is key

Test process



```
Charts charts = new Charts()
// Merge information for the main image in the chart and the secondary images
Map chartInfo = charts.extendChartInfoWithDockerRegistry(
    parseDockerRegistry(clusterInfo.dockerRegistry), opts.chartInfo
)
pushTestingImages(chartInfo, cloud, clusterInfo.dockerconfig, requiresPushToRegistry)

clusterCtx.kubeconfigFile = "${pwd()}/kubeconfig-${cloud}"
writeFile file: clusterCtx.kubeconfigFile, text: clusterInfo.kubeconfig
withEnv(["KUBECONFIG=${clusterCtx.kubeconfigFile}"]) {
    if (shouldExecuteDefaultParametersInstallation) {
        try {
            checkDeploymentWithDefaults(asset, clusterCtx, opts.name, chartPath, chartInfo)
        } catch (e) {
            if (shouldTestBlock(asset, cloud, 'default-parameters')) {
                throw(e)
            } else {
                echo "Installing the chart with default parameters failed. Not blocking releases because of this.\n${e}"
            }
        }
    } else {
        echo "Skipping chart installation with default parameters for '${opts.name}'"
    }
    test(asset, opts.name, clusterCtx, chartPath, chartInfo)
    upgradeTest(asset, opts.name, clusterCtx, chartPath, chartInfo)
}
```

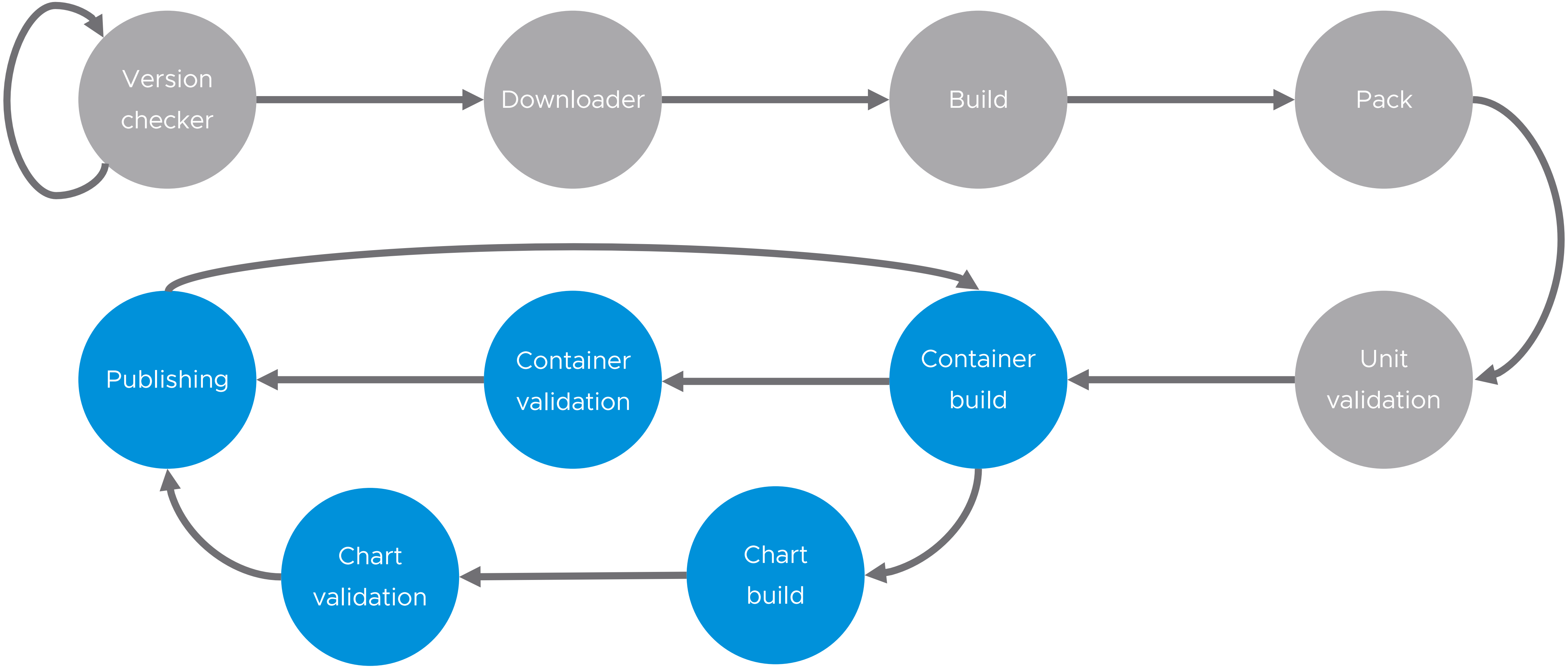
Automation is key

Test process

DEMO TIME

Simplify to go faster

Daily releases



Lessons learned

Security

Update the affected catalog as soon as possible

Code quality

Set up a process to find and fix code issues

Automation

Automate the process as much as you can to have a robust system

Lessons learned

Review your library of tests

Evaluate regularly the tests you perform and balance quality and quantity

Documentation

Generate content for your users and team to follow and keep it up to date

Support

Listen to your users, help them and provide support to all their questions

What’s next?

If you have not tried our Charts yet

Go and get them from https://github.com/bitnami/charts

```
$ helm repo add bitnami https://charts.bitnami.com/bitnami
$ helm install wordpress bitnami/wordpress
```

Check out our tutorials at https://docs.bitnami.com/tutorials

Or visit bitnami.com to deploy any other solution

If you want more from our catalog

VMware Tanzu Application Catalog is a service that combines content and tools to allow IT organizations to consume open source and commercial application customized to their specific requirements, across multiple platforms.

Thank You