Magento®
An Adobe Company

Best Practices Guide:

Developing & Maintaining your Cloud Project

James Doyle, Software Engineer, Magento Services
David Young, Business Solutions Architect, Magento Services

Table of Contents

Magento
An Adobe Company

Introduction

This document assumes the reader has a basic working knowledge of the Magento Commerce Cloud environment and it is intended to convey some suggested practices as well as several common issues of which you should be aware. While all sites are unique and require their own professional assessment prior to modification, these practices are intended to help you have a positive experience while developing and maintaining your cloud-based project.

For a basic overview of Magento Commerce Cloud or a more in-depth development tutorial, please review the following resources:

Dev Docs:

- Welcome to Magento Commerce Cloud

Magento U courses:

- Free Intro to Magento Commerce Cloud
- Magento Commerce Cloud for developers

Magento
An Adobe Company

Git

Integration

Proper usage of the GitHub or Bitbucket integration can be instrumental to the management of your Magento Commerce Cloud project. Configuring this integration allows you to manage your environments directly from your code repository. Using an external GitHub or Bitbucket repository allows you to maintain control of the repository, with admin privileges, and have access to any web GUI or organization level features your vendor provides.

Using the integrations enables you to:

- ✓ Create a new environment when you create a branch
- ✓ Redeploy the environment when you merge a pull request
- ✓ Delete the environment when you delete the branch

The Git repository included with each Magento Commerce Cloud project is primarily intended to manage deployments and is not appropriate for day to day development work, with some exceptions. The only branches that should exist in the Git repository when the integration is initially activated are those from the originally provided Cloud repository. The primary reason for this is that the Cloud system will begin building environments for each branch in the repository being connected. Depending on the age or health of the repository being connected the number of branches can be in the hundreds. Additionally, the branches in the external repository should have identical names matching the original Cloud repository (note that these names are case sensitive).

When activating the integration with your external repository it is recommended to disable 'Build Pull Requests' unless it is specifically needed as it will cause a new environment to be created for each pull request submitted in the connected repository and a re-deployment of said environment to occur any time the pull request has been updated.

Lastly, the user who configures the integration must have admin level access to the external repository to complete this task successfully. See Integrations in the Magento Commerce Cloud documentation.

Branches & The UI

Though each cloud project comes with its own functional Git repository containing a vanilla Magento Commerce Cloud project template it is strongly advised that you use a separate repository for the day to day development work. Due to the architecture of Magento's Cloud environment it's important to understand that each branch in your Git Repository is representative of an environment (active or inactive) and as such reflected in the Cloud UI. While you can have an unlimited number of inactive branches, this is another reason to keep your repository, and thus your project Web UI healthy and clean.

If you're working with a Magento Cloud Pro environment provisioned prior to October 23, 2017 you'll want to ensure that you're utilizing the latest architecture and that both your Production and Staging environments are accessible from the Web UI. If your environment list appears similar to the screenshot below (note the square link to the right of the environment name) then your project should already have the latest architecture in place. If this screenshot is not representative of your project's Web UI you'll just need to complete a few simple steps prior to opening a support ticket to convert your project environment to the most recent version. Please review the materials listed under "Managing Pro Project UI" for a full overview.

Please refer to the following Dev Doc resources on architecture and branches:

Architecture Examples:

- Commerce Pro architecture
- Commerce Starter architecture

Managing Branches & Project UI:

- Project web interface
- Magento CLI
- Managing Pro Project UI

Packages

Cloud Metapackage

The Magento Cloud Metapackage is designed to provide you with the necessary versions of the Magento software as well as the tooling for properly deploying your project to its environments. The process for updating your Magento project's version via composer is intended to be simple, but as the platform evolves, so will the processes. To ensure that your project is configured to retrieve the most up to date versions of all tools while maintaining proper compatibility with your Magento version the Metapackage version should not be hardcoded in your project's composer.json.

For example: `("magento/magento-cloud-metapackage": "2.2.5")` and should not be un-bundled from the Magento core software `(magento/magento-cloud-metapackage)` not present in composer.json

Misconfiguration of this package prevents your project from receiving important automatic updates during deployments. As an example, updating your project to run Magento Cloud 2.2.5 from CLI would be done with the following commands:

```
composer require magento/magento-cloud-metapackage ">=2.2.5 <2.2.6" --no-update
```

```
composer update
```

Note the updated version constraint

Additionally, when upgrading, you'll want to be sure to check in any files marshalled by composer into your repository for deployment as file marshalling is disabled in the Cloud environment.

Please refer to the following Dev Doc resources:

- Upgrading Magento Cloud Version

Magento
An Adobe Company

ECE-Tools

Not only does ECE-Tools provide necessary functionality for effectively deploying your Magento Cloud project with each update it also provides access to hotfixes and patches for your current Magento version. This makes ECE-Tools a fundamental piece of your project that you'll want to make sure is installed and up to date. You can quickly check that it is installed and at what version by running `cat composer.lock | grep "ece-tools"` from the root of the Magento Cloud project in question. To upgrade ECE-Tools to the most recent version the following commands can be executed from the root of the Magento project in your local development environment.

```
composer update magento/ece-tools
```

```
git add -A && git commit -m "Update magento/ece-tools" && git push origin <branch name>
```

Please ensure you have a clean working tree prior to executing the commands as they are written above.

In order to properly leverage the functionality provided by ECE-Tools it's important to verify that all of the necessary deployment hooks are configured in your project's .magento.app.yaml file per the official documentation.

Along with the aforementioned patches and hotfixes Magento strives to provide improvement in performance, stability, and functionality in the tooling as well as the environments those tools support. With the most recent release support was added for Zero Downtime Deployments requiring only minor changes to the static content deployment setting as well as an improved developer experience utilizing Docker to mimic a Cloud like environment locally.

Please refer to the following Dev Doc resources:

- https://devdocs.magento.com/guides/v2.2/cloud/release-notes/cloud-tools.html
- https://devdocs.magento.com/guides/v2.2/cloud/project/ece-tools-update.html
- https://devdocs.magento.com/guides/v2.2/cloud/project/ece-tools-upgrade-project.html#upgrade-the-project

Build Process

Queued Up

The deployment process for all environments in your Magento Commerce Cloud project are handled in a synchronous "first come first served" fashion. All build/deploy requests are processed in the order they are received by the system. Each push to an active branch will trigger a re-deployment of its respective environment and subsequent pushes will not be processed until the active deployment has completed. This can dramatically slow down development speed and is one reason why using an external private repository is advantageous for managing day to day development activities with multiple developers or where code review may be required prior to a deployment being allowed. Unless the system determines that a commit has changed code in the project's repository then the environment will only be redeployed rather than rebuilt and the existing slug will be maintained, more on this in a later section. Based on this information the development workflow for the project should be designed to limit rebuilding environments on an as-needed basis.

One caveat that should be noted: Once a deployment has started and you wish to cancel it or the deployment fails to complete (becomes "stuck" in the queue) it will be necessary to open a Magento support ticket for an agent to intervene.

Please refer to the following Dev Doc resources for additional information on the deployment process:

- https://devdocs.magento.com/guides/v2.2/cloud/reference/discover-deploy.html

SCD Now, Later, or On-Demand

Static Content Deployment (SCD) is a key phase of the build and deploy process as it assembles all the necessary template files, JavaScript, and theme images required to render the frontend of your site. The length of time to complete this process can vary greatly depending on several factors

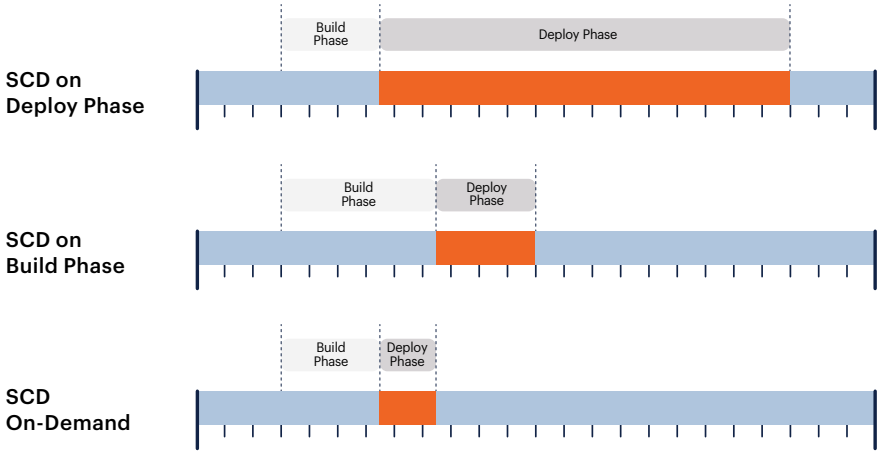
including the number of store views, locales, and themes available as well as the depth of theme inheritance. It's important to understand that while the site is online during the build phase, it is offline during the deploy phase which can have a variable length based on the criteria mentioned above and the configuration options outlined in this section.

SCD is one of the longest and most resource-consuming processes in deploying the store. By default, static content is deployed during the deploy phase, when your site is down—but there are options to improve this:

- SCD on build phase is the functionality that moves static content deployment to the build phase, decreasing downtime and moving heavy processes to a separate build server.
- Alternatively, you may configure SCD on demand in conjunction with a cache warmer to trigger content generation for your site's most important pages, though this is not suitable for all environments.

While generating static content on demand will result in the least amount of downtime while re-deploying your site it should be kept in mind that this configuration option will affect site performance and request throughput until all necessary assets have been re-generated by the system. As noted above, this option should be used in conjunction with a cache warmer to trigger the regeneration of site assets.

Deployment time with SCD on different phases



Please review the following resources on optimizing your SCD:

Dev Docs:
- SCD
- Theme inheritance

Magento Help Center article:
- SCD options to reduce downtime

Imagine 2018:
- Effective implementation on Magento Commerce Cloud

Ideal State Wizard

The Ideal State Wizard is a tool provided as part of the Magento Commerce Cloud deployment toolset (ece-tools) to ensure that your cloud environment is configured correctly based on how you're intending to run certain parts of the platform.

The ideal configuration for your Cloud project helps to minimize deployment downtime by warming the cache and generating static content when requested by the user. If your Cloud is not configured for this ideal state, then you will receive a message similar to the following:

- "SCD on build is not configured"
- "Post-deploy hook is not configured"
- "Skip HTML minification is disabled"
- "Ideal state is not configured"

The ece-tools wizards should be run in your cloud environment from the project's root (e.g. /app). The ideal state wizard can be executed with the following command:

```
./vendor/bin/ece-tools wizard:ideal-state
```

Please review the following Dev Doc resources on this topic:

- Smart wizards
- Release notes for ece-tools

Location of System Logs

The Magento Commerce Cloud server environment may be structured differently than that of another hosting provider, so it's important to know where your project files and service logs are located. Upon connecting to the server via SSH, you will be in the root of your Magento project and your project environment's home directory (referenceable as both `~` and `$HOME`). Depending on the environment you've just connected to the path may differ slightly.

For example: In the integration environment, the path would be `/app`, for Staging it would be `/app/<project_id>_stg/`, and lastly `/app/<project_id>/` for Production.

The standard Magento logs as well as the Cloud specific logs are located relative to the project root (e.g. Logs: /app/var/log and Error Reports: /app/var/report for integration environments) while the system logs remain relative to the system root (/var/log). The specific log files for troubleshooting build and deploy errors in both Pro and Starter accounts are ~/var/log/cloud.log and ~/var/log/install_upgrade.log

Please review the following Dev Doc resource on this topic:

- Project structure (logs)

Redeployment – Quick or Full

Depending on your needs there are two ways you can easily redeploy your Magento Commerce Cloud environment. The 'quick' option is via command line and does not rebuild the entire application prior to the deployment and essentially redeploys the last build pushed to the server. This can be executed with the following command:

```
magento-cloud environment:redeploy –p <PROJECT_ID> -e <ENVIRONMENT>
```

Another option for triggering a quick redeployment without making a change to your codebase would be to push an empty commit into your repository. This can be completed with the following command git commit --allow-empty -m "redeploy" && git push <BRANCH_NAME>. The 'full' redeployment option requires making any change to the codebase then committing and pushing the change. Once a non-empty commit is pushed to an environment branch it will start the build process for the application and subsequently the deployment.

Please review the following Dev Doc resource on this topic:

- Environment commands

Fastly

DNS

Modification of your DNS settings to implement Fastly may seem like a daunting task if you're unfamiliar but it doesn't need to be that way. There are few things you need to ask before making changes:

- ✓ Where does your DNS live?
- ✓ What records are already present?
- ✓ Have you taken a backup of these records prior to making modifications?

You'll find that being prepared and reviewing these items will make the process much easier for you, especially in the event of a mistake or the need to roll back.

After checking with your registrar about where to change your DNS settings, add a CNAME record for your website that points to the Fastly service: prod.magentocloud.map.fastly.net. If you use multiple hostnames for your site, you must add a CNAME record for each one.

Please review the following Dev Doc resource on this topic:

- Fastly DNS

Breaking Full Page Cache (cacheable = false)

This issue is not specific to Magento Commerce Cloud and has the potential to affect all flavors of Magento that are utilizing Full Page Cache. This highlights the necessity for proper code review and testing procedures in your project. In cases where it is necessary to prevent a block from being publicly cached and rendered for all customers, additional steps must be taken to properly implement hole punching and private cache allowing you to specify a time to live (TTL) for the content's expiration and to isolate it to the intended customer. An appropriate time to live set on your private content allows you to maintain fast load times on pages

that are tailored to the customer who's viewing them and the ability to display the most up to date information. The most important thing to understand about this attribute is that it does not just prevent the block it's applied to from being cached but rather prevents the entire page on which the block is contained from being cached which can have major performance implications, especially at scale. Some uses of this functionality are valid, but each instance should be reviewed and assessed by your development team.

The following is an example of a shell command that your development team may use to count the number of times this block attribute appears in your codebase: `ack 'cacheable="false"' | wc –l`

Please review the following resources on this topic:

Dev Docs:

- Cacheable & Uncacheable pages
- Private Content

Video on Mage2.tv:

- Full page caching

Bypassing Fastly for Debugging

Fastly lives as a layer between the customer attempting to view the site and the server responsible for rendering the content where, ideally, we want to keep as much content cached as possible to allow for faster load times and reduced server loads.

In the event you experience some sort of issue in Production or Staging (Fastly is not used in integration) it may be required to bypass Fastly during the troubleshooting process to verify the source of the problem. While investigating whether the issue lies within the underlying build or within cached content that now needs to be purged, we can bypass Fastly and access the origin server directly to validate and determine our next steps. To do this you would need to add an entry to the hosts file on your development computer to point the client's domain directly to the origin server. (STANDARD PATHS-- Linux: /etc/hosts . Windows: C:\Windows\System32\drivers\etc\hosts - Mac: /private/etc/hosts). This local host record entry is a necessary step as the Magento application will redirect any traffic to its configured base_url regardless of the fact the original project hostname or IP are valid at the environment's server level. The following commands can be used to obtain the origin IP of your project's servers based on their region. Any of the IP's returned can be used.

- Magento Commerce Cloud Pro Accounts
  `host <cloud_project_id>.ent.magento.cloud`
- Magento Commerce Cloud Starter Accounts
  `host gw.<region>.magentosite.cloud`
  <region> refers to location of Cloud environments (US-1, US-2, EU, etc.)

Please review the following Dev Doc resource on this topic:

- Fastly troubleshooting

SendGrid

Setup

During the hardware provisioning process the setup for outgoing transactional emails from integration environments is automatically completed for both Starter and Pro level accounts. For the Production and Staging environments additional steps must be taken to complete this setup process and properly configure the DNS to allow outgoing messages to be sent. The DNS records that must be added consist of three (3) CNAME records that resolve to the Domain Keys Identified Mail (DKIM) and Sender Policy Framework (SPF) records managed by SendGrid as well as a unique text (TXT) record which is used for authentication. This allows SendGrid to send transactional emails on behalf of the store as well as reduce the likelihood of these messages being inhibited by spam filters.

Example of DNS CNAME records to be added:

```
em.example.com IN CNAME <sendgrid.net>
s1.example.com IN CNAME <s1.example.sendgrid.net>
s2.example.com IN CNAME <s2.example.sendgrid.net>
```

While this information is covered during the cloud onboarding process is important to ensure that these steps become part of your go-live checklist.

Please review the following Dev Doc & external resources on this topic:

- SendGrid DNS Setup
- DKIM
- SPF

Magento
An Adobe Company

Environment Configuration

Magento
An Adobe Company

Files

Since Magento Commerce Cloud runs in a container-based infrastructure there are several files provided in your project repository that allow you to configure your environment's deployments, services, and the external routes used to reach them.

- **.magento.app.yaml** - defines how to build and deploy Magento, including services, hooks, and cron jobs.
- **.magento.env.yaml** - centralizes the management of build and deploy actions across all of your environments, including Pro Staging and Production, using environment variables. You do not need to open a support ticket to push these changes to Staging and Production environments.
- **.magento/routes.yaml** - configure caching, redirects, and server-side includes.
- **.magento/services.yaml** - defines the services Magento uses by name and version. For example, this file may include versions of MySQL, PHP extensions, Redis, RabbitMQ, and Elasticsearch.

When you push code changes, the active environment provisions container updates using the YAML configuration files. Unlike other YAML configuration files, such as .magento.app.yaml, .magento/routes.yaml, and .magento/services.yaml, you can use the .magento.env.yaml file to manage configuration settings without opening a support ticket.

Please review the following Dev Doc resource on this topic:

- Configure Environments

Support

Magento offers 24x7 support for production critical issues related to your Magento Commerce Cloud infrastructure and core Magento applications not including customizations to the software. The official SLA can be found on the Magento website. As part of your Magento Commerce Cloud Pro account you are provided with both a dedicated account manager to address business related issues as well as a dedicated Technical Account manager to handle technical issues. In the unforeseen event of an outage in a production environment, phone support is provided to address P1 issues. P1 Incidents must be reported on Magento's toll-free support telephone number in order to expedite resolution. Support tickets may be opened via the support portal at **www.support.magento.cloud** for any level of issue. Using the Magento Commerce Cloud status page you can check for issues with any services in your region or subscribe to receive notifications in the event of an issue.

Support ticket portal: www.support.magento.cloud

Status Page: www.status.magento.cloud

P1 Hotline: 1-877-2VARIEN (1-877-282-7436)

P1 Hotline: 001 310-945-1310 (outside the US)

Magento's Service Availability SLA

Service availability is not the same as site uptime

Not included:

- **Magento Customizations**
  Customized Magento application

Included in Magento's 99.99% service availability SLA:

- **Core Magento app environment**
  Server, operating system, databases, php, search, caching, full page cache, and content delivery network
- **Magento's cloud infrastructure**
  Internet network connection and power

Resource Quick Links

1. Introduction

Magento Commerce Cloud

- https://devdocs.magento.com/guides/v2.2/cloud/bk-cloud.html
- https://u.magento.com/magento-commerce-cloud-for-developers
- https://u.magento.com/free-intro-to-magento-commerce-cloud

2. Git

Integration

- https://devdocs.magento.com/guides/v2.2/cloud/reference/git-integration.html
- https://devdocs.magento.com/guides/v2.2/cloud/integrations/bitbucket-integration.html
- https://devdocs.magento.com/guides/v2.2/cloud/integrations/github-integration.html

Branches & The UI

- https://devdocs.magento.com/guides/v2.2/cloud/architecture/cloud-architecture.html
- https://devdocs.magento.com/guides/v2.2/cloud/project/project-webint-branch.html#private
- https://devdocs.magento.com/guides/v2.2/cloud/trouble/pro-env-management.html

3. Packages

Cloud Metapackage

- https://devdocs.magento.com/guides/v2.3/cloud/project/project-upgrade.html

ECE-Tools

- https://devdocs.magento.com/guides/v2.2/cloud/release-notes/cloud-tools.html
- https://devdocs.magento.com/guides/v2.2/cloud/project/ece-tools-update.html
- https://devdocs.magento.com/guides/v2.2/cloud/project/ece-tools-upgrade-project.html#upgrade-the-project

4. Build Process

Queued Up

- https://devdocs.magento.com/guides/v2.2/cloud/reference/discover-deploy.html

SCD Now, Later, or On-Demand

- https://devdocs.magento.com/guides/v2.2/cloud/live/sens-data-over.html
- https://support.magento.com/hc/en-us/articles/360004861194-Static-content-deployment-options-to-reduce-deployment-downtime-on-Cloud
- https://magento.wistia.com/medias/ouh2kncbni

Ideal State Wizard

- https://devdocs.magento.com/guides/v2.2/cloud/env/smart-wizards.html

Location of System Logs

- https://devdocs.magento.com/guides/v2.2/cloud/project/project-start.html#logs

Redeployment – Quick or Full

- https://devdocs.magento.com/guides/v2.2/cloud/reference/cli-ref-topic.html#environment-commands

5. Fastly & FPC

DNS

- https://devdocs.magento.com/guides/v2.2/cloud/access-acct/fastly.html#fastly-dns

Breaking Full Page Cache (cacheable = false)

- https://devdocs.magento.com/guides/v2.3/extension-dev-guide/cache/page-caching.html#cache-over-cacheable
- https://devdocs.magento.com/guides/v2.3/extension-dev-guide/cache/page-caching/private-content.html

Bypassing Fastly for Debugging

- https://devdocs.magento.com/guides/v2.2/cloud/trouble/trouble_fastly.html

6. SendGrid

Setup

- https://devdocs.magento.com/guides/v2.2/cloud/project/sendgrid.html
- https://sendgrid.com/docs/glossary/dkim/
- https://sendgrid.com/docs/glossary/spf/

7. Environment Configuration

Files

- https://devdocs.magento.com/guides/v2.2/cloud/env/environments.html

8. Support

- https://support.magento.cloud
- https://status.magento.cloud/

About Magento Commerce

Magento, an Adobe company, is a leading provider of cloud commerce innovation to merchants and brands across B2C and B2B industries and was recently named a leader in the 2018 Gartner Magic Quadrant for Digital Commerce. In addition to its flagship digital commerce platform, Magento boasts a strong portfolio of cloud-based omnichannel solutions that empower merchants to successfully integrate digital and physical shopping experiences. Magento is the #1 provider to the Internet Retailer Top 1000, the B2B 300 and the Top 500 Guides for Europe and Latin America. Magento is supported by a vast global network of solution and technology partners, a highly active global developer community and the largest eCommerce marketplace for extensions available for download on the Magento Marketplace. More information can be found at **www.magento.com.**

Magento®
An Adobe Company

About Magento Services

Magento Services helps businesses tackle the diverse commerce challenges that face visionary, transformative companies. Multi-disciplinary services provide strategic vision, reinforce quality, optimize performance, and drive operational intelligence. Our edge is derived from our direct access to Magento product owners and engineers, unmatched experience, distinctive end-to-end project approach, and blend of in-house and industry best practices. To learn more about our services, visit our website: **www.magento.com/services.**

Magento® **Services**