

Technical Comparison Sheet: eZ Platform Cloud vs Other Hosting Approaches


This is a technical comparison worksheet between the various approaches to deploy and host your eZ Platform project. Each approach is unique and has its benefits and drawbacks. eZ Platform Cloud is a bundle of the eZ Platform CMS and the PaaS cloud hosting infrastructure.

Since eZ Platform Cloud is provided as a service (PaaS approach) it removes the need of assembling the whole infrastructure yourself. Amazon AWS and Microsoft Azure are Infrastructure as a Service (IaaS) approaches.

They are cloud hosting solutions that offer a range of low level infrastructure systems provided as a service. We then used traditional hosting for different hosting approaches, whether on premises or in a datacenter provided by a third party. A company can either build and purchase or rent a dedicated infrastructure specific to their projects.



Reference Table

 = Out of the box

X = Not provided out of the box

DIY = Do it Yourself (Building or installing)

n.a= Not Applicable

 =More Information

Server Environments

The architecture available out of the box

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
Optimized LAMP Stack		DIY	DIY	DIY
Redundant architecture		DIY	DIY	X
Scaling of the architecture (adding servers, memory, CPUs)	 Upon request but immediate	DIY	DIY	X
24x7x365 monitoring		X	X	X
Deployment workflow based on Git		X	X	X
Dedicated Git Repository		X	X	X
Multiple environments (for dev and test) based on Git branches and pull requests		X	X	X
Assembling solutions based on Docker	X /  Docker is not needed because the purpose of PaaS is to remove the need from building your own infrastructure with technologies such as docker.			DIY

Locations

This will be the data center location of your project. Choose the region closest to the majority of your web traffic, depending on the laws of the country where you're hosting.

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
Available locations	US, Canada, Germany, France, APAC... (most AWS and Azure regions)	US, Canada, Germany, France, APAC...	Numerous	Single Site
Multi-location solutions (application distributed across different servers on different regions/locations)	X	DIY	DIY	DIY

Local Development Tooling

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
Local Development Environment	DIY eZ Launchpad based on Docker or using tunnels	DIY Needs to be assembled from tools such as Docker and deployment tools	DIY Needs to be assembled from tools such as Docker and deployment tools	DIY

Version control

Integration with existing version control tools and services

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
GitHub	✓	n.a.	n.a.	n.a.
BitBucket	✓	n.a.	n.a.	n.a.
Gitlab	✓	n.a.	n.a.	n.a.
Mercurial	✗	n.a.	n.a.	n.a.

PHP, Node.js, Composer

Runtimes Available

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
PHP Versions	5.6x, 7.x	5.6x, 7.x	5.6x, 7.x	5.6x, 7.x
Support for Composer	✓	DIY	DIY	DIY
Support for Node.js	✓	DIY	DIY	DIY
Java	✗	DIY	DIY	DIY
.net	✗	DIY	DIY	DIY

SSL Certificates & HTTPs

Security certificates to encrypt all traffic

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
Automatic Certificate Configuration	✓	✓	X	X
Free Certificates Included	✓	✓	X	X













Caching & CDN

Speed up computation and delivery times with Enterprise-level caching

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
CDN for assets	Cloudfront (EE only)	Cloudfront	Azure CDN (Akami, Verizon, Verizon Premium)	Must be contracted separately
CDN for pages	Fastly (EE only)	Cloudfront (could be done but complex to expire the cache)	Azure CDN (Akami, Verizon, Verizon Premium)	Must be contracted separately
Varnish	✓ Professional plans only	DIY	✓	✓
Local Caching (Nginx, HttpCache, etc)	✓	DIY	DIY	DIY












SSH Access

Access to the filesystem and process running on the remote server

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
SSH Access	 (read-only)			
SFTP/rsync	 By design therefore not neededd-only)			
Root Access	 By design therefore not neededd-only)			

Dedicated Dev/Stage/Prod Environments

A good development cycle requires distinct-but-identical environments for development, staging and testing, and the final Production site.

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
Dev/Stag/Prod environment Parity		DIY	DIY	DIY
Deployment Hooks		DIY (AWS CodeDeploy)	DIY	
Automated Database Management		 (RDS)	 (Azure SQL)	
Automated branching with dedicated environments				

Preconfigured PhpDebug, Blackfire, New Relic Integrations

PhpDebug allows step-by- step debugging for your developers. Blackfire allows you to find performance bottlenecks based on best practice rules you define. New Relic monitors performance in real time.

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
PhpDebug	✔	DIY	DIY	DIY
Blackfire	✔	DIY	DIY	DIY
New Relic	✔	DIY	DIY	DIY

Support & SLAs

A guarantee to stand behind you no matter what, and a promise that your services will stayup and running.

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
Support	Single support service for the whole stack, from hardware to application level	AWS support services only for the low level infrastructure	Azure support services only for the low level infrastructure	DIY
SLA	✔ We have a 99.99% uptime guaranteed for Enterprise plans (Gold++)	✘ Do not have on application because do not know application	✔	✘ (Need to contract)

Backups

Keeping your data safe by saving and storing frequent copies that may be accessed to recover lost information in the unlikely event of an emergency.

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
Backup Solution	✓	✓ (S3/ DB Snapshot-storage container)	✓	✗
Redundant Storage (Multiple copies)	✓ (CEPH)	✓ (Amazon S3)	✓ (Blob Storage)	✗
3rd Party Storage Container Support (Single unit of storage such as Digital Ocean)	✗	✓	✓	✓

Command Line Tooling

Command Line access for management of your environments, projects, branches and code.

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
Environment management from the command line	✓	✓ 3rd party API integrations available	3rd party API integrations available, as well as a rich Microsoft application suite	DIY
Deploy from the CLI	✓	✓	✓	✓
Automatic Tunneling	✓	✗	✗	✗

Business Capabilities

	eZ Platform Cloud	Amazon Web Services (AWS)	Microsoft Azure	Traditional Hosting
Total cost of ownership for the priceline	One single cost for all infrastructure needs	More complex to estimate as different layers of work needs to be estimated	More complex to estimate as different layers of work needs to be estimated	Expensive (Multiple costs)
Time to Deployment	Operational projects in minutes	More complex to estimate as different layers of work needs to be estimated	More complex to estimate as different layers of work needs to be estimated More complex to estimate as different layers of work needs to be estimated	Depending on the project take a lot of time (weeks to months) as you have to do everything for scratch.
SysAdmin & DevOps team	Do not need to assemble and maintain the infrastructure	More complex to estimate as different layers of work needs to be estimated	More complex to estimate as different layers of work needs to be estimated	Requires dedicated team to build and maintain the infrastructure on top of low levels services/bricks
Developer Experience	One environment (cloned from production) available per branch / pull request)	DIY	DIY	DIY
Deployment System	Deployment is abstracted and sync with your own git workflow	DIY (elastic bean stalk) not based on git branch	DIY	DIY

- Professional plans: Based on Platform.sh Professional plans.
- Enterprise plans: Based on Platform.sh Dedicated Enterprise plans.