G-CLOUD 9 FRAMEWORK

SERVICE DEFINITION

PAAS Hosting
platform.sh

Lot 1 Cloud Hosting

ISSUE 1

Service Description

- An overview of the G-Cloud Service (functional, non-functional);

Platform.sh is a ground-breaking hosting and development tool for web applications. It extends a branch-merge workflow to infrastructure so that every branch can be tested as if it were in production, and live sites scale effortlessly to serve the most demanding traffic. Enterprise plans offer a unique, triple-redundant offering, and 24/7 follow-the-sun support. Based on Open Source technologies, Platform.sh is ideal for agile software development, continuous integration and continuous delivery.

With offices in London, the company has technical, support, and sales staff throughout Europe and the United States and provides cloud services to hundreds of companies in various areas including fashion, publishing, retail, travel, engineering and the public sector.

We have over 2,000 customers in 110 countries, including various medium and large (200 sites) UK government implementations.

- Back Up & restore

Automatic snapshots

Snapshots of the production system are automatic, taken every six hours, and retained for 14 days unless otherwise stated.

These are not accessible to the customer and are only used for disaster recovery purposes.

Restore process

Restoration from a snapshot is a managed process done by the Platform.sh team. The periodicity of snapshots can be determined by the customer.

Full site snapshots, comprising codebase, files and database, are run every night. These can be run more regularly if required. These snapshots are stored in individual S3 buckets and are available to customers via a standard support ticket.

- On-boarding processes/scope etc.;

Onboarding procedures
The goals of the onboarding phase are to:

1. Provision the proper Platform Enterprise infrastructure.
2. Provide administrative and developer access to all the necessary resources.
3. Ensure that the customer has the proper concept for migrating and deploying to the Platform Enterprise infrastructure.
4. Support the customer in overcoming any pre-launch issues that arise.
5. Optionally load/performance/security test the site.
6. Ensure the proper functioning of Content Delivery Network (CDN), Email, and any

other supporting systems that are part of Platform.sh.

7. Explain the DNS requirements, guide the customer through a successful switch of DNS and launch of the site.

These goals are achieved using the following process:

1. Welcome Email: introducing the customer to Platform.sh, their Account Manager (for commercial topics and escalations) and contact information for the Support team.
2. Kick-off call: here the customer team (commercial and technical) meets the PLatform.sh team and together they complete the onboarding document, adding project timelines, technical requirements, migration details and any issues/risks
3. Follow-up calls are also scheduled to tackle specific topics like DNS, CDN, SSL and integrations
4. Load, performance and security tests are conducted as required
5. During the pre-launch phase the support team provides ongoing assistance in migrating and preparing the site for go-live.
6. A launch window is planned and support resources made available to accompany this.

- Off-boarding processes/scope etc.;

Offboarding procedures

Should the customer for reasons of non-performance and semi-permanent or regular failure terminate the contract, Platform.sh will refund 70% of the remaining balance, calculated as at the end of the final month during which service is being delivered. This allows us to cover the cost of reservation deposits.

Customers wishing to terminate service can have all their data returned to them upon request.

- Service management details;

Maintenance windows

Our policy regarding system maintenance and downtime is to provide notification one week in advance of the scheduled outage in the form of an email which provides the following information:

- when the outage will occur and its duration (date and time zones EST/CET)
- who is affected by the outage (which clusters/customers)
- how the outage is affecting them (eg: Access to Platform.sh UI temporarily unavailable if the user has not already logged in.)
- why the outage is occurring

There may also be a banner on the Platform.sh website to notify customers who may not have seen the email.

Level of customisation permitted

Platform.sh Enterprise can be customized and extended in many ways, e.g. by adding

services, PHP extension or integrations. This is done based on presales discovery during which the Platform.sh technical team assesses the technical viability and fit.

Platform.sh Standard exists as a single predefined package, but allows for considerable customization. Customers can define their complete infrastructure and configuration using simple .yaml files and choose from a range of services including PHP 5.4, 5.5, 5.6, Hip Hip VM, Maria DB, Mongo, Solr, ElasticSearch, Kafka, Rabbit MQ and others. The Platform.sh roadmap foresees the integration of many other services including NodeJS, Python, Ruby and Java. Platform.sh allows customers to add their own php.ini files. Customers can also define build-time services such as Grunt, Sass, Bower and Gulp.

Schedule for deprecation of functionality

Platform.sh supports the current version(s) of the software in our stack. In the run up to a software version being retired, the Platform.sh team will work with customers to plan for a successful migration to newer versions, e.g. PHP 5.4 to 5.5 or 5.6. Customers who remain on older versions of software will be required to commit to a migration roadmap in which upgrades are planned for the short to medium term.

Platform.sh is particularly well suited to this kind of upgrade and customers can simply run dedicated branches to compare functionality in existing and new version of the software.

Service Levels (e.g. performance, availability, support hours, severity definitions etc.);

Platform.sh Enterprise offers a monthly uptime percentage SLA of 99.99% on production infrastructure.

1. Definitions

- “Monthly Uptime Percentage” means the percentage derived by subtracting from 100 the percentage of Service Unavailability minutes during the month.

- “Service Unavailable” means the inability of the Platform to run web applications, due either to errors or failures in the hosting stack or lack of network connectivity to the Internet.
- “Service Credit” means a dollar credit, calculated as set forth below, that Platform.sh may credit back to an eligible account.
- “Supported Users” mean the authorized users of Platform Services (“Supported Users”) listed by Subscriber on Addendum B, specifically, the contact information for at least one and a maximum of three representatives who will be authorized to submit tickets and exercise rights for hosting granted by this Agreement. No other Subscriber representatives will be so entitled, and they will be redirected via email to one or more of the Supported Users. Subscriber will promptly notify Platform.sh in writing of any changes to the Supported Users.

2. Service Commitments and Service Credits

- Service Commitments. Platform.sh agrees to host and maintain Subscriber’s websites

and associated development environments on the Platform for the term of the Agreement and in accordance with the service level standards set out below. Platform.sh shall use commercially reasonable efforts to prevent unauthorized access to its infrastructure and shall promptly notify Subscriber of any known security breaches.

- Service Credits. Service Credits are calculated as a percentage of the total charges paid for Platform use by the affected live sites for the monthly billing cycle in which the Service Unavailable status occurred, in accordance with the schedule below.

Monthly Uptime Percentage	Service Credit Percentage
Less than 99.99% but equal to or greater than 99.95%	5%
Less than 99.95% but equal to or greater than 99.0%	10%
Less than 99.0%	30%

Platform.sh will apply any Service Credits only against future Platform payments otherwise due. Platform.sh may at its discretion issue the Service Credit to the credit card used by the Subscriber to pay for the billing cycle during which there was Service Unavailable. Service Credits will not entitle Subscriber to a refund or other payment. A Service Credit will be applicable and issued only if the credit amount for the applicable monthly billing cycle is greater than one dollar (£1 USD). Service Credits may not be transferred or applied to any other account. Unless otherwise specified in writing, Subscriber's sole and exclusive remedy for any unavailability, non-performance, or other failure by Platform.sh to provide Platform-related Services is the receipt of an eligible Service Credit.

Support hours

Priority 1 tickets are covered 24/7/365. All other support will be performed during standard UK business hours, 9:00 AM to 6:00 PM.

Severity definitions

- Priority 1 (P1) - P1 is a catastrophic production problem that severely impacts the Subscriber's production systems, or because of which Subscriber's production systems are down or not functioning, or that results in a loss of production data and no workaround exists. Platform.sh will make continuous efforts, with appropriate escalation to senior management, to provide a resolution.

- Priority 2 (P2) - P2 is a problem in which the Subscriber's system is functioning but in a reduced capacity, or the problem is causing significant impact to portions of business operations and productivity, or the software is exposed to potential loss or interruption of service. Platform.sh will make efforts during business hours to provide a resolution.

- Priority 3 (P3) - P3 is a medium- to low-impact problem that involves partial and/or non-critical loss of functionality, or that impairs some operations but allows the Subscriber's operations to continue to function. Problems for which there is limited or no loss of functionality or impact to the Subscriber's operation and for which is there is an easy workaround qualify as P3. Platform.sh will use reasonable efforts to provide a resolution in time for the next minor release of the software.

Support Request Methods

In the event of an emergency, customers can submit a support ticket to us through several different methods:

- Submit a ticket through your https://accounts.platform.sh/user account with a Priority of *Urgent*:
- *A short summary of the problem the user is facing*
- *A short overview of the steps required to re-create the issue*
- *Any related screenshots or error messages*
- Submit a ticket to us from the Platform.sh UI by clicking on the "Support" menu in the upper right corner of any Platform.sh project page. Use the Submit ticket menu item to navigate to the Platform.sh support form and provide the same information requested in bullet 1) above.

Response Times Depending on your subscription, the response time will vary:

- Platform.sh Enterprise customers:
- *P1 tickets: 2hrs or less from reception of your support request in the ticket system.*
- *P2 tickets: 4hrs or less from reception of your support request in the ticket system.*

- *P3 tickets: best effort*
- Platform.sh Standard customers: Best effort for all cases and priority levels

- Training;

As outlined above, Platform.sh provides comprehensive onboarding services. Detailed documentation is available at docs.platformsh, and we also offer a live chat service where customers can be given real-time assistance on technical and commercial questions.

- Ordering and invoicing process;

Describe process for ordering services from organisation.

For Platform.sh Professional

- Currently available sizes are Small, Medium, Large, X-Large and XX-Large
  - These sizes will accommodate a range of <300k to <6m pageviews per month ● There is an additional cost to add more development users, storage and environments
  - Sizing is the responsibility of the customer. Sales can be contacted for advice.
  - These can be procured through self-service subscription via the web-site.
  - Credit Card is required.
  - Invoicing on request, contact sales.

For Platform.sh Enterprise

- Currently available sizes are PE6, PE12, PE24, PE48, PE96, PE192, PE384 and bespoke
  - These sizes will accommodate a range of <2m to 100m+ uncached page views per month
  - There is an additional cost to add more than the included development users, storage, environments. Bandwidth usage over the 1Tb per month included can be charged pre-paid in advance or post-paid in arrears.
  - There is also an additional cost for dedicated IP addressing, staging servers, CDN option (AWS Cloudfront is included in the price), extra docroots, extra web servers
  - These are procured through the sales organisation.
  - Sizing is undertaken by Platform.sh sales on receipt of questionnaire https://platform.sh/quote/. If you unsure about answering any of the questions, please contact sales.
  - These are procured via our sales team, and require a specific contract and invoiced payment.

- Termination terms:

If not terminated the contract will renew for a further year from the last day of the hosting term. Should the customer for reasons of non-performance and semi-permanent or regular failure terminate the contract, Platform.sh will refund 70% of the remaining balance, calculated as at the end of the final month during which service is being delivered

- Data processing & storage location

Platform.sh ensures that customer confidential data stays on AWS or Azure infrastructure and is only held outside of AWS or Azure for short periods of time when this is necessary in order to solve an issue or debug a problem connected with the data.

Customers can choose where they want to have their data stored. For UK clients the preferred locations can be any datacentre they choose. Once a customer has selected a region Platform.sh undertakes to hold that customer's data only in the region specified.

- Data restoration / service migration;

Recovery from Failure on Platform.sh Enterprise

Since everything is triple-redundant on Platform.sh Enterprise, you can easily simulate failure by disconnecting one of the three servers from the cluster. This is an important operation that the Platform.sh team regularly tests, and Customer is welcome to participate in this test on their sites. Each Platform.sh Enterprise cluster is designed to withstand the loss of an entire server and all of the services running on it.

Automatic snapshots

Snapshots of the production system are automatic, taken every six hours, and retained for 14 days unless otherwise stated.

These are not accessible to the customer and are only used for disaster recovery purposes.

Restore process

Restoration from a snapshot is a managed process done by the Platform.sh team. The periodicity of snapshots can be determined by the customer.
Full site snapshots, comprising codebase, files and database, are run every night. These can be run more regularly if required. These snapshots are stored in individual S3 buckets and are available to customers via a standard support ticket.

- Customer responsibilities;

Customers are expected to have both commercial and technical staff available to interface with the Platform.sh team and any third parties involved in the project, such as agencies. During critical phases such as onboarding, the customer should plan to have resources available to ensure that this process goes smoothly.

- Technical requirements

Our website and support offering is available in English only. We do not currently provide any specific disability capabilities.

- Browsers

The Platform.sh GUI is accessible from any modern web browser including Chrome, Safari, Opera, Firefox and Internet Explorer

- Details of any trial service available.

We offer a free one-month trial for Platform.sh Standard with no further commitment required. After one month users can either terminate their trial or convert to a paid package.

- Where the G-Cloud Services do not include a feature listed in this Schedule the Service Definition should include clearly that the feature is not included.

We are introducing new products in 2017 (Local Development) that are not currently specified or priced.

Information assurance/Security

Data-in-transit protection

All sites on Platform.sh Enterprise must use SSL. New or existing SSL certificates can be installed on Platform.sh Enterprise via a standard support ticket.
By default, key management is done by Platform.sh's own identity and authentication management database. Only protocols supporting strong encryption are used for data transfer (https, scp, sftp, rsync). Upon request (add-on module) a FIPS-compliant enterprise key manager will be deployed to govern key access. This also opens the door to encrypted data at rest (at the volume level or individual database fields).

Asset protection and resilience

Platform.sh offers a comprehensive set of measures in order to protect customer assets and provide excellent system resilience: **Read-only architecture**

As a result of Platform.sh Enterprise's unique architecture, all services running in production are mounted on read-only partitions. Read/write mount points (e.g. files in the database) are explicitly identified, and easily auditable; and no program located there can be executed. This makes the majority of attacks very difficult.

Service/process isolation

Processes in Platform are totally separated, with each process running in a separate LXC container (exactly as if each process running on a separate physical server). It is extremely difficult (if not impossible) to use a flaw in one component to attack another.

Code-driven infrastructure

Platform.sh builds the complete server environment on each Git commit, with services (e.g. PHP, MySQL, Solr etc.) defined in simple YAML files that reside in the Git repository. As a result, the infrastructure is kept to the absolute functional minimum, with no unnecessary services or daemons running.

Network isolation

Platform.sh Enterprise uses a cluster dedicated to each virtual network. It is impossible to connect from one cluster to another. IP services are on a non-routable range, so it is impossible to connect to a container other than the "Gateway". The gateway only permits three services: HTTP, HTTPS and SSH (ports 80, 443, and 22). Thus each cluster is in a DMZ beyond a double Firewall (with Amazon protecting the Gateway, and the Gateway protecting the clusters).

Servers are only accessible via SSH and git/ssh access to the servers. SSH users access by adding their public key via the Platform.sh UI. Access rights are highly granular and assigned per environment.

Virtualization Security

Platform.sh utilizes tightly managed IAM Users, Groups, and Roles at the AWS/Azure level to govern user access. VM images are built and maintained by the Platform.sh team. OS-level Access is governed by key-based access and keys are managed in our central authentication database. Active data at rest (the data being used by the application) can be encrypted upon request (add-on module). Backup data at rest is encrypted by default.

Physical Security

All Platform.sh Enterprise resources run in Amazon Web Services or Azure facilities. These data centers are state of the art, utilizing innovative architectural and engineering approaches. Amazon has many years of experience in designing, constructing, and operating large-scale data centers. This experience has been applied to the AWS/Azure platform and infrastructure. AWS data centers are housed in nondescript facilities. Physical access is strictly controlled both at the perimeter and at building ingress points by professional security staff utilizing video surveillance, intrusion detection systems, and other electronic means. Authorized staff must pass two-factor authentication a minimum of two times to access data center floors. All visitors and contractors are required to present identification and are signed in and continually escorted by authorized staff.

Systems Access Controls System Access and Authentication

Identity and authentication is managed by a central database which enforces Two Factor Authentication (TFA) for Platform.sh personnel. TFA can be mandated by customer for customer access. The protocol used is OAUTH2 + OpenID Connect. An additional layer of access is governed by public/private key pairs; any developer wishing to push code to Git or

open an SSH session must possess the proper private key.

Physical access

AWS/Azure only provides data center access and information to employees and contractors who have a legitimate business need for such privileges. When an employee no longer has a business need for these privileges, his or her access is immediately revoked, even if they continue to be an employee of these companies. All physical access to data centers by employees is logged and audited routinely.

Stack patch management

Platform.sh subscribes to all relevant security lists for Platform.sh Enterprise elements. The complete stack is kept up to date with all security updates via the Debian security channel.

Platform.sh maintains their own apt-get repository. No updates are included without vetting and approval by Platform.sh.

Antivirus

If required, for example where customers or site users are uploading files to the server, customers can request to have ClamAV installed on their cluster via a standard ticket.

ClamAV is an open source antivirus engine for detecting trojans, viruses, malware & other malicious threats.

Static Security Checks

On deployment Platform runs static security checks on some highly critical vulnerabilities and will fail the deployment if those are included. The security checks are signature based. For example if a user pushes a Drupal installation that is vulnerable to CVE-2014-3704 the deployment will stop and the developer notified.

These are not end-to-end tests but designed not to interfere with development while catching some of the most dangerous and widespread vulnerabilities. When a new highly critical vulnerability is known the platform team proactively adds its signature.

Separation between consumers

Service/process isolation

Processes in Platform are totally separated, with each process running in a separate LXC container (exactly as if each process running on a separate physical server). It is extremely difficult (if not impossible) to use a flaw in one component to attack another.

Network isolation

Platform.sh Enterprise uses a cluster dedicated to each virtual network. It is impossible to connect from one cluster to another. IP services are on a non-routable range, so it is impossible to connect to a container other than the "Gateway". The gateway only permits three services: HTTP, HTTPS and SSH (ports 80, 443, and 22). Thus each cluster is in a DMZ beyond a double Firewall (with Amazon protecting the Gateway, and the Gateway protecting the clusters).

Servers are only accessible via SSH and git/ssh access to the servers. SSH users access by

adding their public key via the Platform.sh UI. Access rights are highly granular and assigned per environment.

Governance

Security incident response

Platform.sh subscribes to all relevant security lists for Platform.sh Enterprise elements. The complete stack is kept up to date with all security updates via the Debian security channel. Once a vulnerability has been detected, a standardised process is followed:

- the Security Team meets to identify all clients affected;
- the order in which patches will be applied is prioritised based on need;
- customers are proactively informed of the upcoming patch via a ticket;
- the patch is applied to a staging environment and tested;
- Once the patched site has passed all requisite tests the patch is applied to the live environment and the customer is informed of the successful patching. Should a customer report a vulnerability the same process is also followed.

Security for customer confidential information

APPLICATION

All operations are automated in Platform.sh Enterprise. Individual instances and clusters are brought online and connected programmatically, without the need for a system administrator to connect each machine to its destination cluster. This reduces the scope for human error, the leading cause of security flaws.

All requests to the system are logged and stored in a dedicated S3 bucket.
Cluster-internal instances log applications locally as standard. Customers can request access to logs via a standard ticket.

Furthermore:

- The Platform.sh team never accesses customer data directly
- Options exist for encrypting data at rest: volume encryption and database field encryption - Data is encrypted in transit via HTTPS/SSL

OPERATIONAL

Platform.sh logs all operational activity, centralises log management, basing operational alerts and monitoring on automated triggers. These use Munin, Nagios, Logstash, Kibana, and Elasticsearch among others.

EMPLOYEE ACCESS TO CUSTOMER INFORMATION

Only a very small and select group of direct employees on the operations team has any access to the customer database.

Data is never accessed directly by the service operations team, nor is it moved, for any reason, without prior consent. However, the subset of super admins who are at the highest privilege level for operations would theoretically have visibility on stored data. For example, a super admin on the server on which BC data is stored would be able to see file names of images uploaded to Drupal, and is able to exert administrator level operations on them. Highest privilege sysadmins are also able to initiate database connections to the BC database. This is a usual situation with the operating environment.

Our offering is based on a standard set of processes and controls to enable skilled staff the access they require in the maintenance of application and hosting provision. We take security very seriously and staff undergo a rigorous selection process, and are continuously monitored and trained. Senior management review processes and spot check adherence.

Configuration and change management

System maintenance and downtime

System maintenance is occasionally required to improve or correct issues involving the platform. They do not, however, systematically require downtime for our customers, and in some cases this can be avoided if the customer remains logged in to their Platform.sh UI.

Downtime is avoided as much as possible, and when it is unavoidable we prioritize scheduling it during off-business hours for EST/GMT/CET time zones.

Scheduled maintenance

Our policy regarding system maintenance and downtime is to provide notification one week in advance of the scheduled outage in the form of an email which provides the following information:

- when the outage will occur and its duration (date and time zones EST/CET)
- who is affected by the outage (which clusters/customers)
- how the outage is affecting them (eg: Access to Platform.sh UI temporarily unavailable if the user has not already logged in.)
- why the outage is occurring

There may also be a banner on the Platform.sh website to notify customers who may not have seen the email.

Vulnerability management

On deployment Platform runs static security checks on some highly critical vulnerabilities and will fail the deployment if those are included. The security checks are signature based. For example if a user pushes a Drupal installation that is vulnerable to CVE-2014-3704 the deployment will stop and the developer notified.
These are not end-to-end tests but designed not to interfere with development while catching some of the most dangerous and widespread vulnerabilities. When a new highly critical vulnerability is known the platform team proactively adds its signature.

Incident response

Platform.sh subscribes to all relevant security lists for Platform.sh Enterprise elements. The complete stack is kept up to date with all security updates via the Debian security channel. Once a vulnerability has been detected, a standardised process is followed:

- the Security Team meets to identify all clients affected;
- the order in which patches will be applied is prioritised based on need;
- customers are proactively informed of the upcoming patch via a ticket;

- the patch is applied to a staging environment and tested;
- once the patched site has passed all requisite tests the patch is applied to the live environment and the customer is informed of the successful patching. Should a customer report a vulnerability the same process is also followed.

Event monitoring

Incident response
Platform.sh subscribes to all relevant security lists for Platform.sh Enterprise elements. The complete stack is kept up to date with all security updates via the Debian security channel. Once a vulnerability has been detected, a standardised process is followed:

- the Security Team meets to identify all clients affected;
- the order in which patches will be applied is prioritised based on need;
- customers are proactively informed of the upcoming patch via a ticket;
- the patch is applied to a staging environment and tested;
- once the patched site has passed all requisite tests the patch is applied to the live environment and the customer is informed of the successful patching. Should a customer report a vulnerability the same process is also followed.

Incident management

Incident response

Platform.sh subscribes to all relevant security lists for Platform.sh Enterprise elements. The complete stack is kept up to date with all security updates via the Debian security channel. Once a vulnerability has been detected, a standardised process is followed:

- the Security Team meets to identify all clients affected;
- the order in which patches will be applied is prioritised based on need;
- customers are proactively informed of the upcoming patch via a ticket;
- the patch is applied to a staging environment and tested;
- once the patched site has passed all requisite tests the patch is applied to the live environment and the customer is informed of the successful patching. Should a customer report a vulnerability the same process is also followed.

Personnel security

Our staff are hired based on technical excellence and domain knowledge; we also assess the candidate's character and career path, disqualifying unsuitable applicants of questionable standing or moral character.

Only a very small and select group of direct employees on the operations team has any access to customer data.

Data is never accessed directly by the service operations team, nor is it moved, for any reason, without prior consent. However, the subset of super admins who are at the highest privilege level for operations would theoretically have visibility on stored data. For example, a super admin on the server on which data is stored would be able to see file names of images uploaded to Drupal, and is able to exert administrator level operations on them. Highest privilege sysadmins are also able to initiate database connections to the database. This is a usual situation with the operating environment.

Our offering is based on a standard set of processes and controls to enable skilled staff the access they require in the maintenance of application and hosting provision. We take security very seriously and staff undergo a rigorous selection process, and are continuously monitored and trained. Senior management review processes and spot check adherence.

Secure development

Platform.sh allows customers to automatically sanitise DBs and effectively restrict developers to their own branches so that they can pull changes from upstream and make pull requests but not merge into upstream branches without intervention from a team member with higher permissions.

Supply-chain security

Platform.sh runs on AWS/Azure and leverages their mature and comprehensive security models including world-leading supply chain management.

Platform.sh itself is produced 100% internally and does not require external supply chains outside the organisation.

Authentication of consumers

We provide access to our services via the following methods:

- SSH
- Oauth - Google, Github etc
- Strong username/password

Separation and access control within management interfaces

Platform.sh provides complete segregation of user accounts and data leveraging several systems including the Drupal CMS and Git.

Client logos

Case Studies or use cases

https://platform.sh/files/en/platformsh-british-council-case-study.pdf - British Council global Drupal multisite global migration to containers

https://platform.sh/files/en/Platform.sh%20-%20Reiss-Testamonial.pdf - Reiss, global high street eCommerce fashion retailer

Contact details

Kieron.sambrook-smith@platform.sh, +44(0)7818 427136