Qubell

Qubell Adaptive Platform-as-a-Service, Enterprise Edition

Architecture Overview

4600 Bohannon Drive, Menlo Park, CA 94025 **T** 888 855-8940 http://qubell.com

Introduction

Qubell Adaptive Platform-as-a-Service provides enterprises with capabilities to ensure efficient development, deployment, testing, releasing and rolling back software projects on heterogeneous, geo-distributed infrastructure. Qubell Platform integrates with development tools, such as version control systems and build servers, with configuration management tools such as Puppet and Chef, and with IaaS (Infrastructure-as-a-Service) solutions such as Amazon EC2. OpenStack and VMWare vCloud to provide end-to-end application management solution.

Qubell Platform is available in two editions, Express and Enterprise. The following information applies to Qubell Adaptive Platform-as-a-Service, Enterprise edition.

Qubell Platform is developed, operated and supported by Qubell, Inc., a Delaware company.

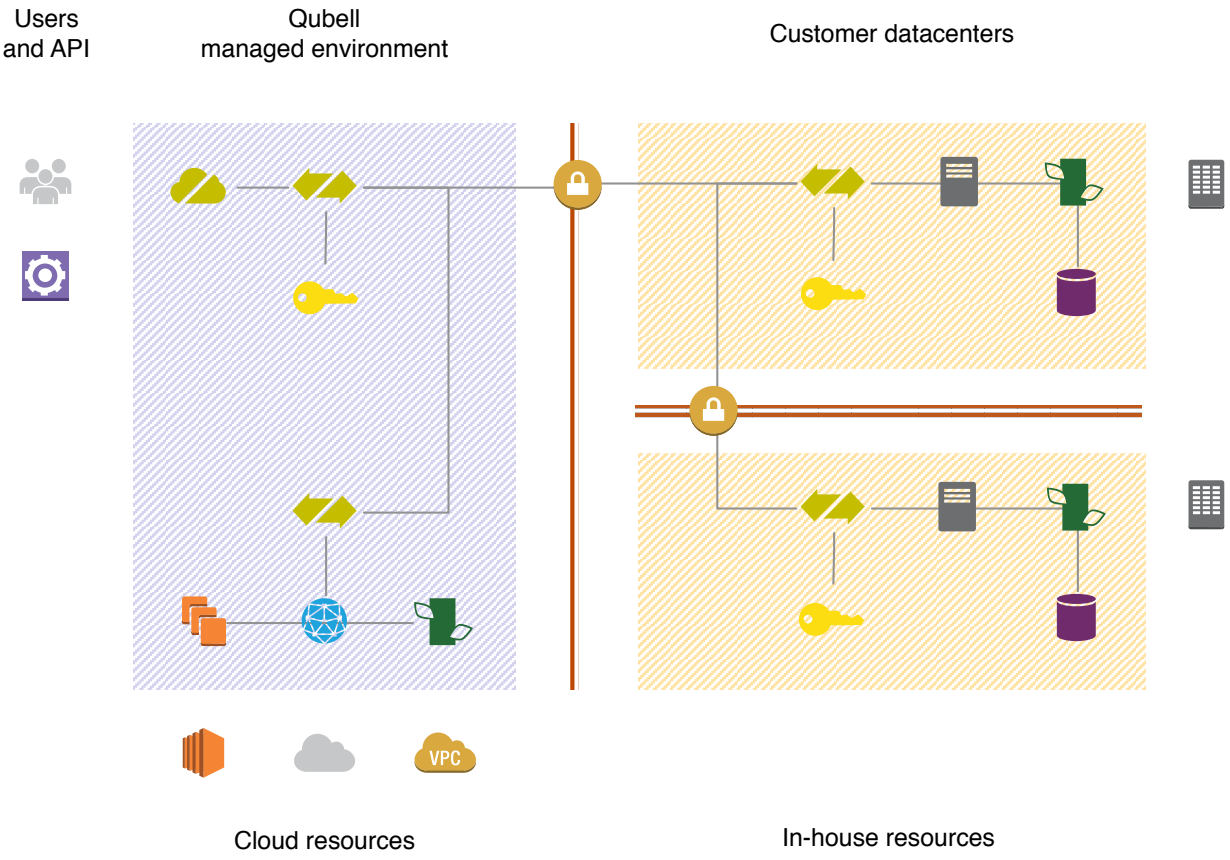
Table of Contents

Platform Overview

Qubell Platform prescribes a model-driven approach to application management. An application architect willing to use Qubell Platform needs to define a schema, known as application manifest. The manifest is then registered with Qubell Platform, where it is verified and stored in the database. When a new application instance is requested, a manifest is compiled into a binary form, which is then used by the system to provision resources, deploy, configure, scale, test, monitor and destroy the application instance.

While some applications can be operated in isolation, most of them are supposed to be operated in some kind of application execution environment. Qubell Platform allows users to define execution environments as a set of services accessible to the application and policies that govern application configuration. This provides a way to codify operational differences within the environment definition, while maintaining a unified application manifest. An example of such differences would be production vs non-production service endpoints, performance profiles, or data sets.

In terms of physical architecture, Qubell Platform is a hybrid deployment system. It consists of: (1) Collaboration Portal, a centralized management interface, delivered as SaaS and operated by Qubell and (2) Control Fabric, a network of independent controllers, managed and operated by the jointly by Qubell and the customer. This allows the fabric to extend behind the corporate firewall and provide unified system controls while satisfying rigorous security and compliance policies.



Generalized architecture diagram of Qubell Adaptive Platform-as-a-Service. Users and external systems interact with the platform by talking to the Collaboration Portal on the left, which in turn communicates with fabric controllers deployed in different data centers. A network of fabric controllers is jointly known as Control Fabric, which manages applications and mediates access to resources, both on-premise and in the cloud.

The following sections will explain the inner workings and guarantees of Collaboration Portal and Control Fabric separately.

Collaboration Portal

Overview

Collaboration portal manages the business processes related to schema, asset and configuration management. Core use cases implemented by the collaboration portal are described below.

1. Schema authoring: upload a an application or service manifest; register an externally hosted manifest.
2. Policy authoring: add, remove or modify environment policies and associated services.
3. Command-and-control: launch and destroy application and service instances, monitor instance status, execute scaling and testing workflows.
4. Change management: define application configurations and orchestrate configuration changes.
5. Notification center: configuring and receiving notifications about system events.
6. Time-based scheduling: auto-destroy an instance after a specified time interval.

Data

Collaboration portal does not store highly sensitive data (e. g. passwords) or personally identifiable information.

Collaboration portal processes and stores the following data:

1. Application schemas. Special data files, called manifests, specify logical components of the application and dependencies between them. They are created by engineering teams and always stored and processed on the collaboration portal.
2. Service definitions. Similar to application schema, service manifests, created by the operations teams, specify the endpoints for the shared services that are consumed by the running applications.
3. Configuration data. Specific configuration items that are instrumental to the application behavior are kept on the portal. This may include OS version, location of specific libraries installed on the server or connection string to management interfaces.
4. Environment policies. Environment policies are rules that define how the application should behave during deployment and operation. They codify the differences between same application been deployed production and development purposes.

5. Runtime metadata. Information about the state of each application and service instance, IP addresses of the running virtual machines, application URLs and usage information are sent to the portal by the fabric for aggregation and reporting.

6. Authentication and authorization data. If using built-in user directory for authentication, user passwords are hashed with bcrypt and stored in the database. Authorization data includes identities, user IDs, roles, and permissions that comprise access controls to the portal, which in turn provides access to the control fabric.

7. User profiles and notifications. User profiles include primary email address and miscellaneous setting related to the user interface.

Perimeter

Perimeter of the portal is composed of three distinct interfaces: user and application programming, gateway and management.

User and application programming interfaces

Collaboration portal provides the users with an ability to access stored data and perform various command-and-control operations, including launching and altering the state of running application instances. These can be performed either through a web UI or a REST-based API for the purposes of integration with automated tools. Both options utilize the same code path and provide same security guarantees.

All communication must happen over HTTPS, using at least SSL v3. Accessing UI and API over plain HTTP or using SSL version less than 3 is not supported.

Authentication is handled by one of three supported protocols [1]:

1. Password-based authentication with cookie-based session. Password are verified against a strong bcrypt hash stored in the user database.

2. Basic HTTP authentication over SSL (API only).

3. OpenID-based authentication against a trusted provider, with a cookie-based session. Currently only Google Apps is supported as a trusted provider.

Authorization is handled by a Role-based Access Control (RBAC) module. Implementation of RBAC in Qubell Platform allows granting the roles with granular permissions to individual applications, environments and instances.

Fabric gateway

Gateway is used as an endpoint for the fabric controllers. This interface is used by the fabric to address following use cases:

1. To collect and aggregate data generated by the fabric.

2. To communicate change in schemas, configuration or environment policies.

3. To transmit user command to the fabric.

---

[1] Qubell is working on a fourth method of authentication, using SAML identities provider by the corporate identity provider (e. g. Microsoft Active Directory).

To achieve this, portal exposes four endpoints that are used by the fabric controllers:

1. Two active sockets (symmetric, for HA purposes).

2. Two standby sockets used as BCP targets (symmetric, for HA purposes).

Controllers must be registered at the portal and provide valid identity token to be able to connect to the gateway.

Selection of the active socket and failover to a second active socket is performed automatically by the controller. Failover to a BCP target and back does not happen automatically and must be initiated by the operator.

All communication is done over a single multiplexed connection initiated by the controller, through a proprietary binary protocol over SSL v3 and above.

IP addresses and ports of all gateway endpoints can be fixed by customer's request, making it possible to limit the range of IP addresses that can be accessed by the controller.

Management interface

Management interface is used only by Qubell staff and is not customer-serviceable. For the outline of Qubell security practices, refer to the Appendix.

Control Fabric

Overview

Qubell Control Fabric is a network of connected agents (fabric controllers) that work together to reach the goals set up by the portal (infrastructure provisioning, application deployment, testing, scaling, etc.) in the context of the environment policies.

Every controller defines a control zone, which is a collection of resources and services consumed by the applications. Access to the services is mediated through environment policies. Control zone usually corresponds to a datacenter or availability zone within a datacenter. Control zones can comprise a two-level hierarchy, with the topmost zone connecting to the portal gateway.

Fabric controller consists of several loosely-coupled services, which are responsible for different aspects of controller behavior. The two most important services are compute interface and secure vault.

Compute interface provisions virtual Linux-based servers using a selected cloud provider and executes shell or Chef scripts on the servers through SSH. It uses password-less keys by default and can be used under a non-privileged user. Chosen privileged commands can be executed through password-less sudo invocation, enabling system administrators tight control over script execution.

Secure vault stores and processes highly sensitive data such as keys and passwords.

Data

Every fabric controller stores and processes the following data:

1. Runtime metadata. Information about the state of each application and service instance, IP addresses of the running virtual machines, application URLs and usage information are stored in the database of the controller of the appropriate zone.
   In addition, a controller that is designated as router will route (but not store) the same kind of data to and from other zones without storing it.

2. SSH keys and service access credentials. Highly sensitive data is kept in a Secure Vault, a specific module of the controller. This data is encrypted with zone key using a symmetrical AES cypher both when at rest and in transit, making it unusable outside the zone.

3. Connection information. Each controller keeps a list of endpoints and tokens that are used for connecting to an upstream controller.

4. Zone key. The controller uses this key to encrypt highly sensitive data within the zone. The same key can be shared between different zones to enable sharing of highly sensitive data.

Perimeter

Perimeter of a controller is composed of four distinct interfaces: upstream, downstream (gateway), user and management.

Upstream

All controllers connect to the collaboration portal, either directly or through a gateway of another controller.

Controllers must be registered at the portal and intermediary controllers and provide valid identity token to be able to connect to the gateways.

All upstream communication is done over a single multiplexed connection initiated by the downstream controller, through a proprietary binary protocol over SSL v3 and above.

IP addresses and ports of all gateway endpoints can be fixed by customer's request, making it possible to limit the range of IP addresses that can be accessed by the controller.

Fabric gateway

Downstream (gateway) interface must only be enabled on the intermediary controllers that route the data across the zones. Gateway interface can be considered a server interface for the downstream controllers (clients).

Downstream interface consists of two active sockets (symmetric, for HA purposes). All downstream communication is done over a single multiplexed connection initiated by the downstream controller, through a proprietary binary protocol over SSL v3 and above.

IP addresses and ports of all gateway endpoints can be fixed by customer's request, making it possible to limit the range of IP addresses that can be accessed by the downstream controller.

User interface

Controller user interface is playing a supporting role to the portal user interface. The only use case for the user interface on the controller is key upload/download, described below.

Portal user interface provides a graphical user interface to upload and download SSH keys that are used by the controller to invoke commands on servers in the control zone. However, the portal does not process highly sensitive data such as SSH keys. Instead, it instructs the appropriate controller to generate a unique one-time link that is used for key upload and download. The actual transmission of keys then happens over a direct secure channel with an appropriate controller. If not used, the link expires within a minute. This makes it impossible to download the keys outside of the firewall.

Controller user interface is a web-based interface, delivered over HTTPS, using at least SSL v3.

Management

Access to management interface differs between whether the controller in question is managed by Qubell (managed scenario) or by the customer (unmanaged scenario).

Managed scenario

Management interface is used only by Qubell staff and is not customer-serviceable. For the outline of Qubell security practices, refer to the Appendix.

Unmanaged scenario

In the unmanaged scenario, management interface is used by the customer staff to perform following activities:

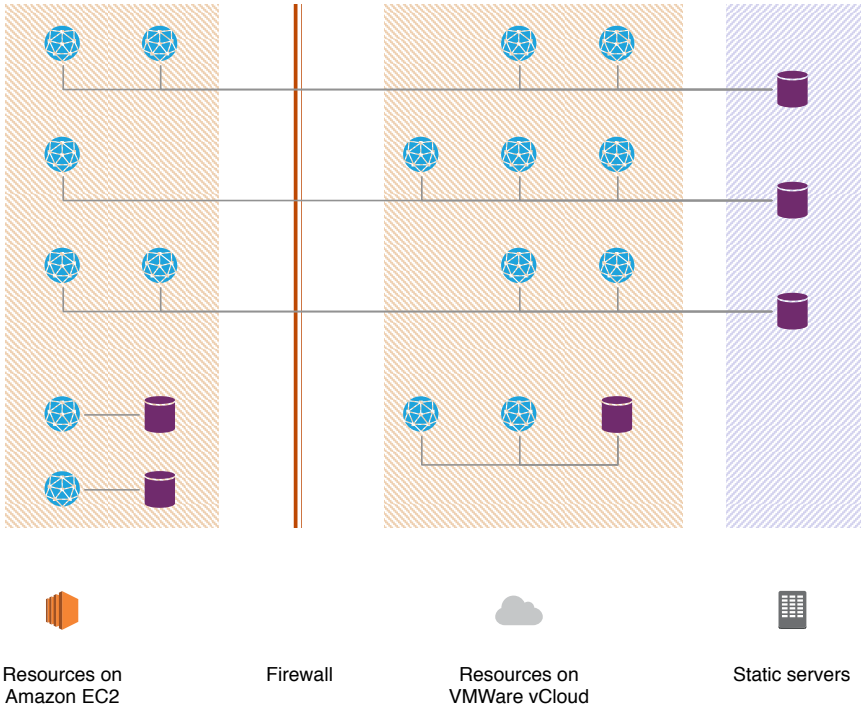
1. Installation and upgrade. Controllers are distributed as RPM packages and require Red Hat Enterprise Linux or CentOS 6. Qubell provides updates releases of the controllers on a regular basis. Such releases might contain important security updates.
2. High availability. Qubell provides tools and reference architectures to help customers ensure high availability of the controller.
3. Monitoring. Qubell provides tools and reference architectures to integrate controller monitoring into the enterprise monitoring system, as well as basic lights-out management.
4. Log retention. Qubell provides the location of all the logs generated by the controller.
5. Backups. Qubell provides backup and restore instructions for the data stored at the controller.
6. OS and server management.

For the details on operating the controller, see the Agent Operation Guide.

Sample Deployment

For the purposes of this example, we will consider the following requirements to the solution:

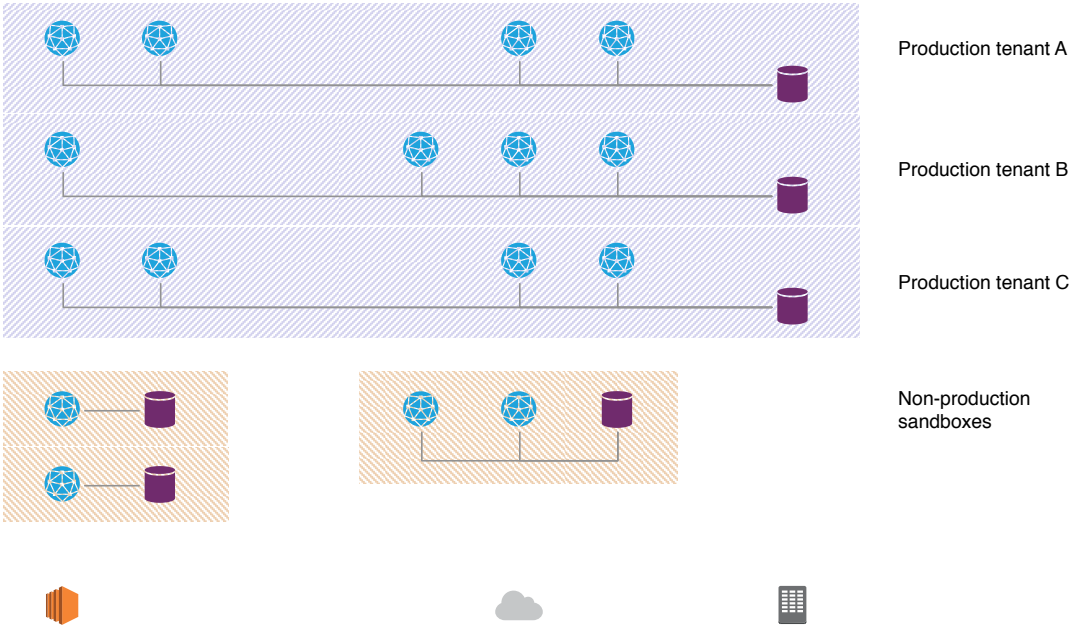
1. Support production deployment of multiple independent instances of a web application across two data centers.
2. Support database servers running on static hardware.
3. Support web servers running on vCloud and Amazon Web Services.
4. Support deployment of smaller, fully virtualized non-production instances for development and test purposes.



Resources (web and database servers) spread over three physical domains.

Logically, these resources are aggregated into several instances the same application, deployed for production or non-production use in different configurations.

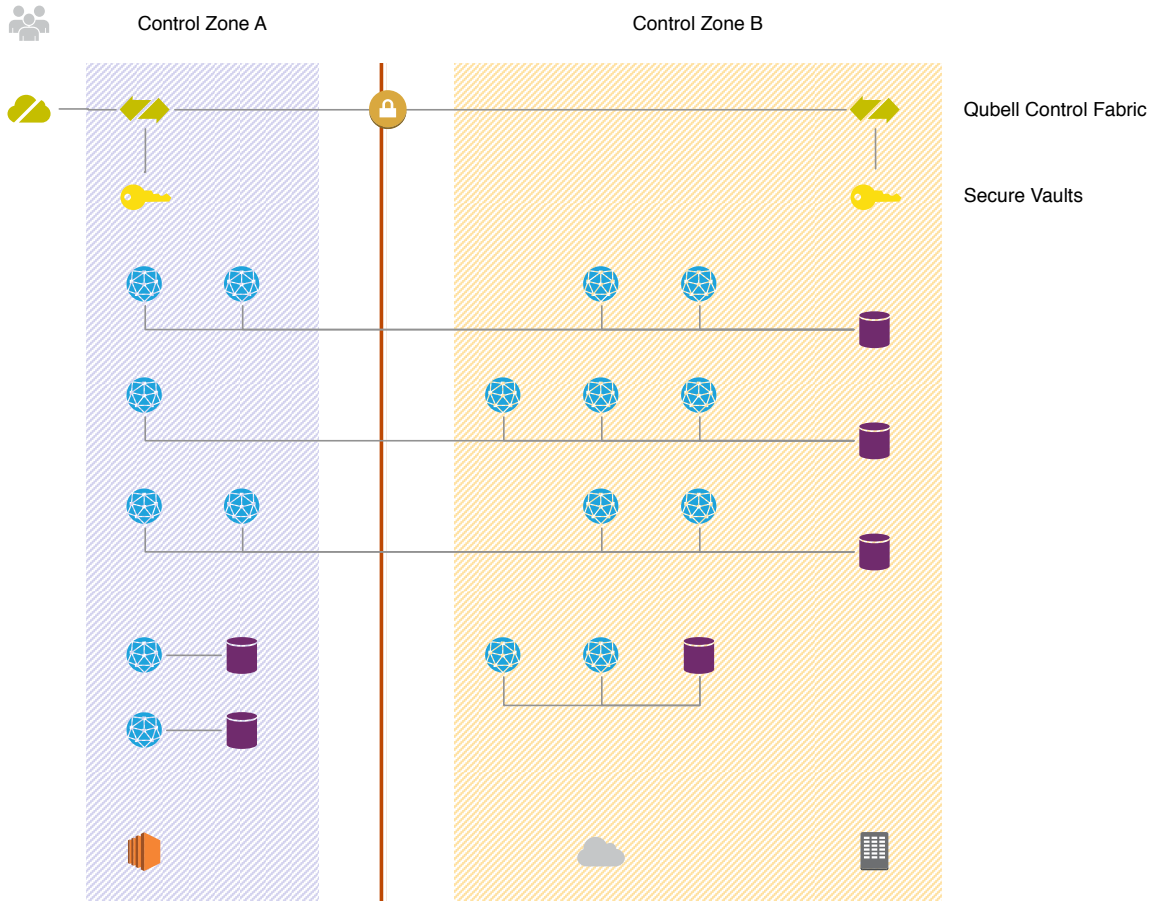
Qubell provides a single application view for all instances, both individual production tenants or non-production sandboxes:



Resources grouped together according to the dependencies to form application instances.

To enable management of the distributed applications, Qubell Platform is deployed in the following configuration:

1. One unmanaged Fabric Controller (Controller A) is deployed behind the firewall. It connects to the local resources (vCloud Director, virtual machines, static servers) and manages creation of the behind-the-firewall resources for application instances requested by the users. Secure Vault A contains the keys necessary to perform behind-the-firewall operations. Keys are encrypted with the machine key of Controller A and cannot be encrypted outside of Control Zone A.
2. One managed Fabric Controller (Controller B) is deployed at the AWS region. It connects to AWS resources (API and virtual machines), manages creation of AWS resources and acts as an upstream router for the controller A. Secure Vault B contains the keys necessary to perform AWS operations. Keys are with the machine key of Controller A and cannot be encrypted outside of Control Zone B.
3. Portal is configured to route user requests through the controller B.



Multi-zone deployment of Qubell Control Fabric.

Appendix: Qubell Environment

The following information applies to managed components of Qubell Adaptive Platform-as-a-Service, Enterprise Edition.

Compliance

Qubell is refining its internal processes to become achieve ISO 27001:2005 certification in Q1 2014.

Availability

All of the critical components of the system are mirrored by a hot standby for high availability. In the event of a failure of one of the components, failover is 100% automated. All core services of Qubell Platform are covered by 99.5% availability SLA.

Qubell maintains a cold disaster recovery target in a different geographic region. In case of a disastrous outage, operation is restored within 24 hours, with no more than 4 hours of data loss.

Qubell is providing each customer with a unique status page that helps tracking availability and maintenance events.

Backups

All of the customer data is backed up off-site every 4 hours and retained for 30 days. Data can be restored from a certain date by customer request.

Security

All of the customer data is stored and processed in SSIE16-compliant facilities.

Each customer receives a dedicated database with unique access credentials. Processing of customer data happens on dedicated virtual servers.

All servers undergo regular vulnerability scanning with a prompt assessment and resolution of any discovered vulnerabilities.

Qubell retains 30 days of security logs for audit purposes.

Updates

Backwards-compatible updates are announced beforehand and applied to the managed servers during regular maintenance windows. Potentially disruptive updates are scheduled with customer in advance.

Support

Qubell provides a 8/5 technical support for questions and issues, plus a 24/7 support for security issues.

Glossary

Amazon Elastic Compute Cloud
IaaS offering from Amazon, part of *Amazon Web Services*. Provides an ability to rent virtual servers by the hour.

Amazon Web Services
Set of related IaaS offerings from Amazon, see http://aws.amazon.com/.

AMQP
See *Advanced Message Queuing Protocol*.

Advanced Message Queuing Protocol
Open specification addressing bidirectional flow of messages over a TCP connection. See http://www.amqp.org/.

Controller
(special usage) See *Fabric Controller*.

AMQPS
AMQP over TLS. See *AMQP*, *TLS*.

API
See *Application Programming Interface*.

Application Programming Interface
A well-defined interface used by third-party applications to consume the capabilities of the platform.

Application execution environment
A set of policies regulating application deployment across *control zones*, its operation, and *resources* and *services* accessible to the application.

Application instance
A single instance of an application running within an *application execution environment*.

Application manifest
A machine-readable model of the application structure, defined by the *managed application components* and dependencies between such components.

Application revision
A configuration of application, comprised of application manifest, version of code, data, configuration files and related artifacts that should be eventually released to production.

bcrypt
Secure hashing function based on Blowfish algorithm.

Chef
Popular open-source application configuration management system. See http://www.opscode.com/chef/.

Cobalt
(special usage) Internal code name for the *Control Fabric*. Can be occasionally seen in the configuration files. See *Control Fabric*.

Component
See *Managed application component*.

Control Fabric
A federation of control zones, managed by a network of *fabric controllers*.

Control zone
A set of managed *services* and *application components* controlled by a single agent.

Secure Vault
A specialized service that is responsible for providing credentials to the consumers within the *control zone*.

EC2
See *Amazon Elastic Compute Cloud*.

Environment
See *Application execution environment*.

Fabric Controller
A stand-alone system that is responsible for controlling of managed *services* and *application components* within a single *control zone*.

Fabric
(special usage) See Control Fabric.

Git
A popular open-source, distributed source control system. See http://git-scm.com/.

GitHub
Proprietary SaaS or shrink-wrap software product that is based on *git*. See http://github.com/.

IaaS
See *Infrastructure-as-a-Service*.

Infrastructure-as-a-Service
A category of *SaaS* offerings designed to provide virtual compute, storage and network resources, usually metered and billed by the hour.

Instance
(special usage) See *Application instance*.

Jenkins
Popular open-source build server. See http://jenkins-ci.org/.

Managed application component
A well-defined piece of an application that has to be managed on its own, as opposed to being an integral part of an application.

Manifest
See *Application manifest*.

OpenStack
Popular open-source *IaaS* framework. See http://www.openstack.org/.

OpSource
IaaS provider in California, now part of Dimension Data Holdings. See http://www.opsource.net/.

Puppet
Popular open-source application configuration management system. See http://puppetlabs.com/puppet/puppet-open-source/.

RBAC
See *Role-based Access Control*.

Role-based Access Control
A NIST (www.nist.gov) model of access control based on the concept of logical roles and associated permissions. Implementation of RBAC in Qubell Platform allows granting the roles with granular permissions to individual applications, environments and instances.

REST
See *Representational State Transfer*.

Representational State Transfer
A style of software architecture and design for Web services.

SaaS
See *Software-as-a-Service*.

Service
(special usage) See *Zone service*.

Software-as-a-Service
A style of software delivery based on subscription-based access to the software over the Internet.

SPI
See *Service Programming Interface*.

Service Programming Interface
A well-defined interface used by service developers to extend the capabilities of a platform.

TLS
See *Transport Layer Security*.

Transport Layer Security
Cryptographic protocols that provide communication security over the Internet. See http://tools.ietf.org/html/rfc5246.

Qubell Adaptive Platform-as-a-Service, Enterprise Edition
Edition of Qubell Platform targeted at enterprise customers. Includes strengthened SLAs and support for behind-the-firewall execution zones.

Qubell Adaptive Platform-as-a-Service, Express edition
Edition of Qubell Platform targeted at individuals and small businesses. Only public *IaaS* providers are supported in Express version.

Collaboration portal
User-facing tier of Qubell Platform; includes API for integration purposes.

vCloud
Popular *IaaS* product from VMWare. See http://vcloud.vmware.com/.

Zone service
Extension to the core platform capabilities that is accessible within a *control zone* and communicates with the *fabric* through the *SPI*.

Zone
See Control zone.