

What's New in CA API Gateway 9.1?

CA Communities Webcast
April 26, 2016

Aaron Flint, Senior Product Manager
Hirbod (Rod) Moshfeghi, Senior Product Manager



Agenda

1

Overview

2

Docker Gateway

3

Silent Gateway Install and Configuration

4

Developer License

5

Developer UX enhancements

6

Gateway Performance enhancements

7

Other new features

8

Q&A

Overview – CA API Gateway 9.1

Available now (GA release April 22, 2016)

- **New Docker-based Gateway**

- Docker form factor for Gateway, Silent install and configuration enhancements

- **Developer Workstation**

- Developer workstation form factor, CentOS based virtual appliance

- **Developer UX enhancements**

- Publish Swagger services in Policy Manager, Policy Manager task menu re-organization, improved audit functionality

- **Gateway performance improvements**

- Policy Manager login speed improvements, CA Siteminder auth cache, replay protection functionality enhancements

- **Other enhancements**

- Support for cloud-init, re-introduction of Thales HSM support

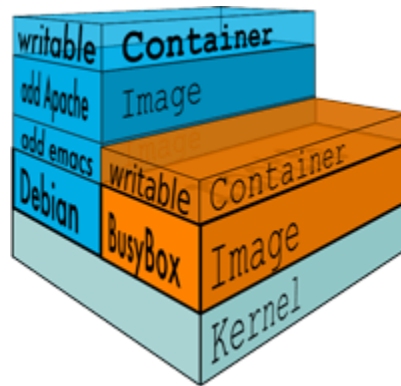
CA API Gateway - Docker Container

Overview of Docker

What is Docker?

“Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, system libraries – anything you can install on a server”

-<https://www.docker.com/what-docker>

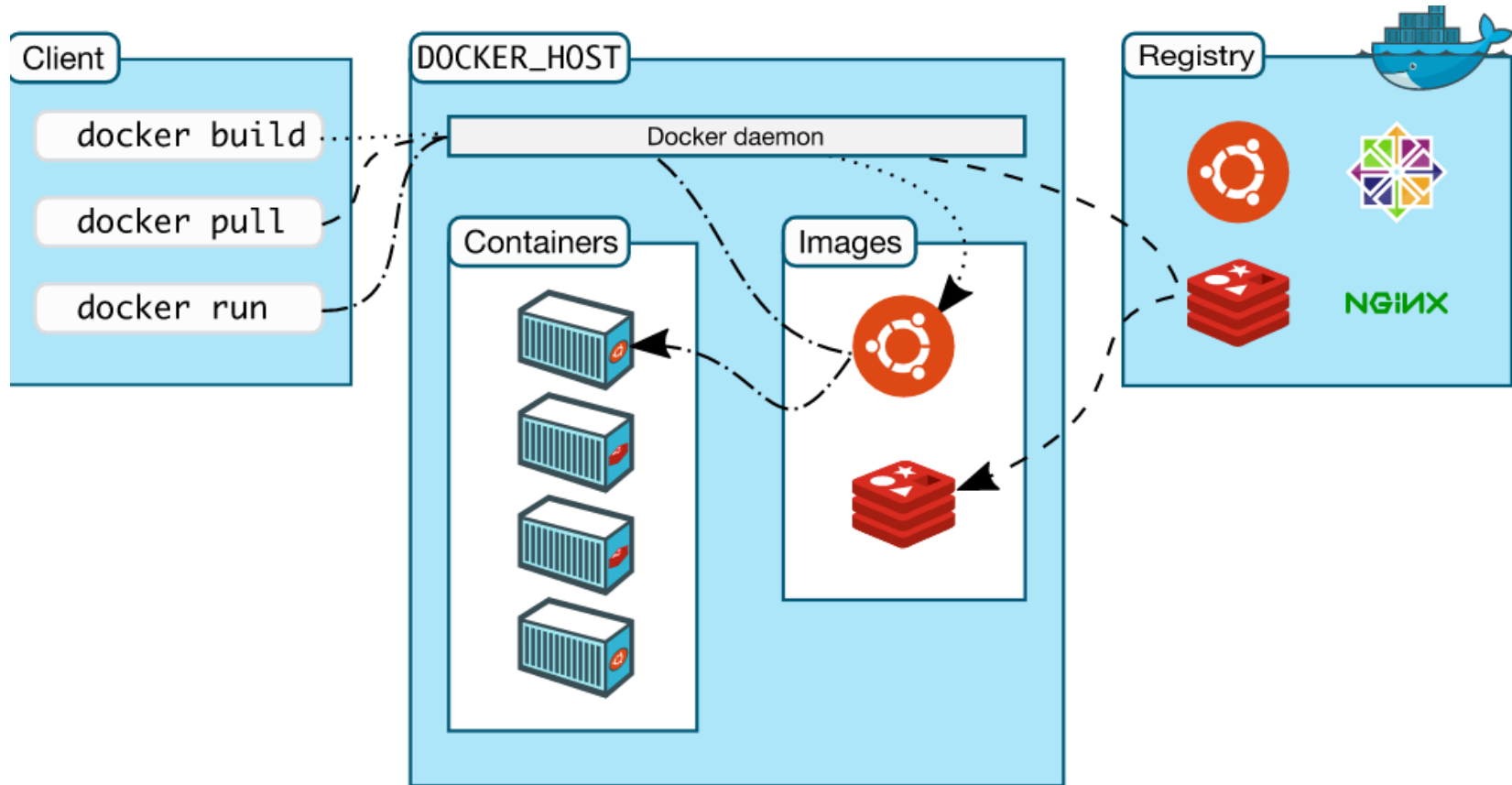


Why use Docker?

- Lightweight
- Portable: build once, run anywhere
- Improve Development Velocity
- Burgeoning Eco system

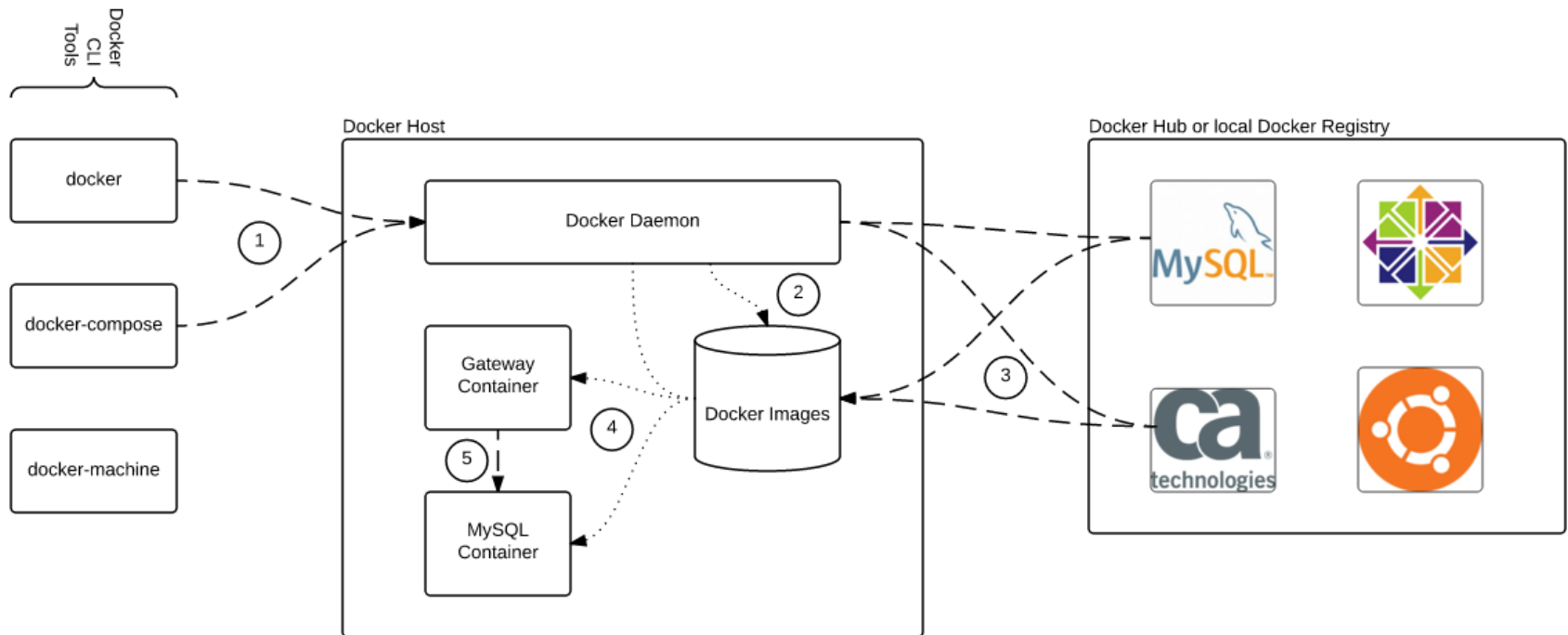
CA API Gateway - Docker Container

Docker Deployment Model



CA API Gateway - Docker Container

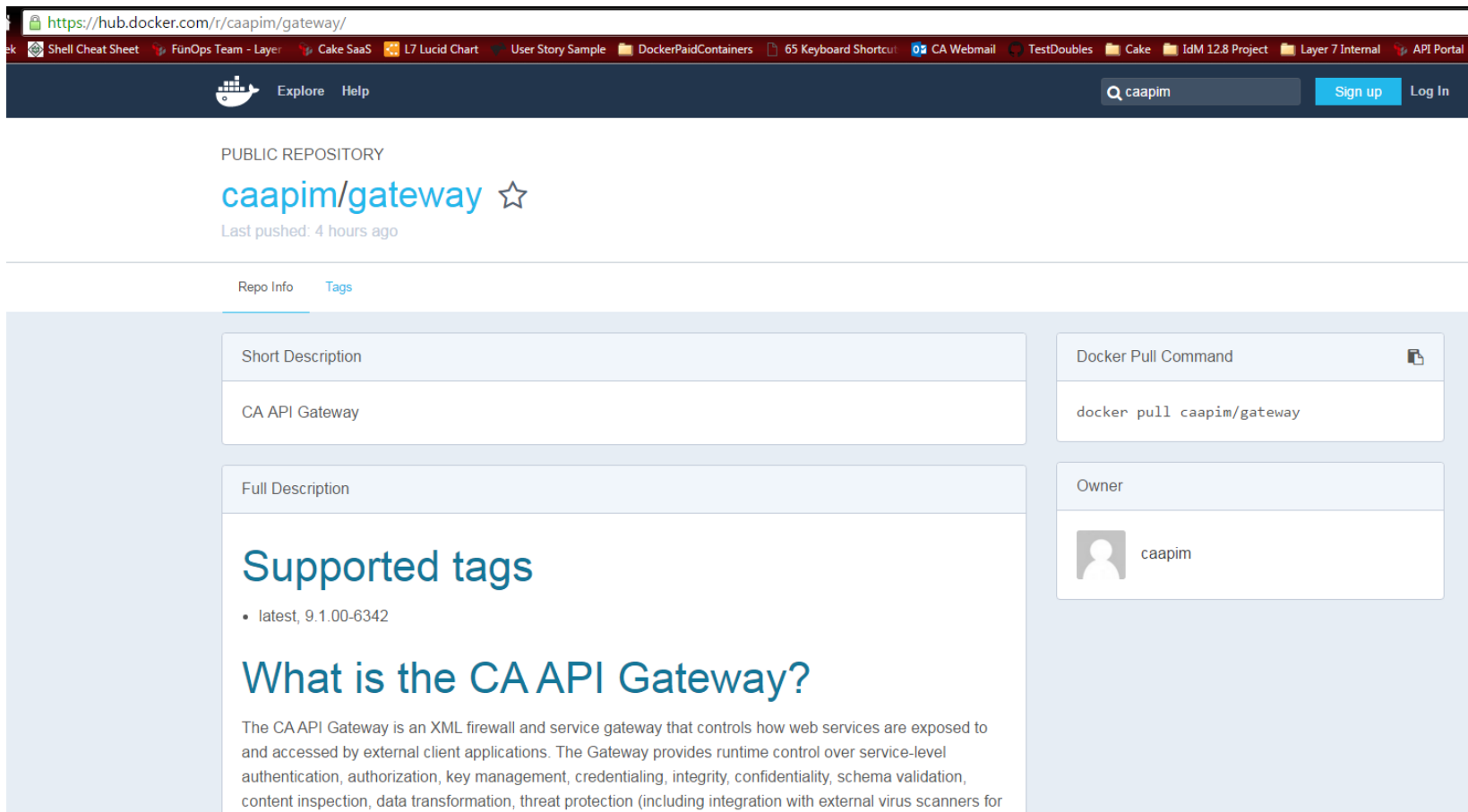
Gateway Docker Deployment Model



CA API Gateway - Docker Container

What does CA provide?

Docker Container, via DockerHub



The screenshot shows the Docker Hub interface for the `caapim/gateway` repository. The page is titled "PUBLIC REPOSITORY" and shows the repository name `caapim/gateway` with a star icon. Below the name, it says "Last pushed: 4 hours ago". The page has tabs for "Repo Info" and "Tags". The "Repo Info" tab is active, showing a "Short Description" of "CA API Gateway" and a "Full Description" section. The "Full Description" section includes a heading "Supported tags" with a list of tags: `latest`, `9.1.00-6342`. Below this, there is a heading "What is the CA API Gateway?" followed by a paragraph: "The CA API Gateway is an XML firewall and service gateway that controls how web services are exposed to and accessed by external client applications. The Gateway provides runtime control over service-level authentication, authorization, key management, credentialing, integrity, confidentiality, schema validation, content inspection, data transformation, threat protection (including integration with external virus scanners for". To the right of the description, there is a "Docker Pull Command" section showing the command `docker pull caapim/gateway` and an "Owner" section showing the user `caapim`.

https://hub.docker.com/r/caapim/gateway/

Shell Cheat Sheet FunOps Team - Layer Cake SaaS L7 Lucid Chart User Story Sample DockerPaidContainers 65 Keyboard Shortcut CA Webmail TestDoubles Cake IdM 12.8 Project Layer 7 Internal API Portal

Explore Help

caapim Sign up Log In

PUBLIC REPOSITORY

caapim/gateway ☆

Last pushed: 4 hours ago

Repo Info Tags

Short Description

CA API Gateway

Full Description

Supported tags

- latest, 9.1.00-6342

What is the CA API Gateway?

The CA API Gateway is an XML firewall and service gateway that controls how web services are exposed to and accessed by external client applications. The Gateway provides runtime control over service-level authentication, authorization, key management, credentialing, integrity, confidentiality, schema validation, content inspection, data transformation, threat protection (including integration with external virus scanners for

Docker Pull Command

```
docker pull caapim/gateway
```

Owner

caapim

CA API Gateway - Docker Container

What does CA provide?

Getting Started Guide

Sample docker-compose.yml files, for easy deployment

CA API Gateway - 9.1

Documentation powered by DocOps

Search for...

PDF EPUB Versions Language

- Using the Gateway Docker Appliance

Docker: Deployment Model

Set Up Local Environment for Docker

Verify that Docker Image can be Pulled

Prepare Docker Compose

Connect to a Docker Gateway using the Policy

Manager

Appendixes: Gateway Docker

Appliance

± Configure Security

- + Publish Services and Configure Policies

- + Policy Assertions

- + Context Variables
- + Cluster Properties

- Upgrade, Migrate, Patch, Back Up,

- + Administer the Gateway

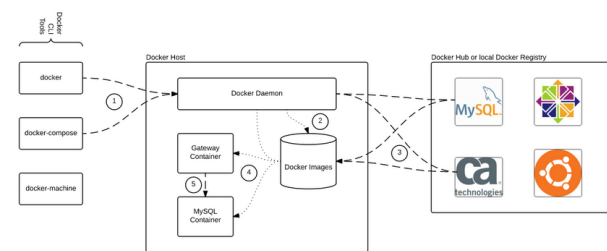
[Contents](#) |
 [Home](#) /
 [Install and Configure the Gateway](#) /
 [Using the Gateway Docker Appliance](#)

Docker: Deployment Model

Last update April 21, 2016

This topic describes a sample deployment of the CA API Gateway using Docker CLI tools. Other deployments may be possible using other tools and schedulers in the Docker ecosystem (for example, Kubernetes or OpenShift 3).

The following diagram illustrates the example Docker deployment:



```

1  ssg:
2    image: caapim/gateway:9.1.00-6342
3    mem_limit: 2048m
4    expose:
5      - "8777"
6    ports:
7      - "8080:8080"
8      - "8443:8443"
9      - "9443:9443"
10   environment:
11     SSG_CLUSTER_COMMAND: "create"
12     SSG_CLUSTER_HOST: "localhost"
13     SSG_CLUSTER_PASSWORD: "7layer"
14     SSG_DATABASE_TYPE: "mysql"
15     SSG_DATABASE_HOST: "mysql"
16     SSG_DATABASE_PORT: "3306"
17     SSG_DATABASE_NAME: "ssg"
18     SSG_DATABASE_USER: "gateway"
19     SSG_DATABASE_PASSWORD: "7layer"
20     SSG_DATABASE_ADMIN_USER: "root"
21     SSG_DATABASE_ADMIN_PASS: "7layer"
22     SSG_ADMIN_USER: "pmdadmin"
23     SSG_ADMIN_PASS: "7layer"
24     SSG_LICENSE:
25       SSG_INTERNAL_SERVICES: "restman wsman"
26   links:
27     - mysql
28   mysql:
29     image: mysql:5.5
30     mem_limit: 512m
31     # volumes:
32     #   - ./mysqlconf:/etc/mysql/conf.d
33     #   - /root/mysqldata:/var/lib/mysql
34   environment:
35     - MYSQL_ROOT_PASSWORD=7layer


```


CA API Gateway - Docker Container

Customer Collaboration

Alpha Release on
validate.ca.com

[←](#) [→](#) [C](#) [H](#) [https://validate.ca.com/project/version/item.html?cap=59EDA46CFB2945F5A559E8232C7B43F5&arttypeid={B6F6D8E6-6F9C-...}](#) [Apps](#) [Sarek](#) [Shell Cheat Sheet](#) [FunOps Team - Layer](#) [Cake SaaS](#) [L7 Lucid Chart](#) [User Story Sample](#) [DockerPaidContainers](#) [65 Keyboard Shortcuts](#)

 Customer Validation Community

[Centercode Support](#)

[Home](#) [CA APIM](#) [Alpha Releases](#) [API Gateway 9.1 Docker Alpha](#)

Project Management

- [Project Tools](#)
- [Users](#)
- [Reports](#)

Feedback

- [Defect Reports](#) [SUBMIT](#)
- [Suggestions](#) [SUBMIT](#)
- [Questions](#) [SUBMIT](#)

Resources

- [User Forums](#)
- [Project News](#)
- [Agreements](#)
- [Charters](#)
- [Reviews \(recordings\)](#)
- [Alpha Releases](#)

API Gateway 9.1 Docker Alpha

Overview

If you have reached this page, thank you for interest in the API Gateway Docker Container. In the spirit of collaboration, we would like to share an early version of our Container with you to get feedback on our new offering.

Pre Requisites

- A Docker Hub Account:
 - You will need this to pull our Docker Container from our Docker Hub Repository
- Somewhere to run Docker
 - A machine of some sort will do. You can even use your laptop! The attached "Getting Started Guide" will help you setup the Docker Tools.

Getting Started

1. Download the artifacts listed in the "Downloads" Section
2. Contact **Hirbod (Rod) Moshfeghi** (hirbod.moshfeghi@ca.com) and give him your Docker Hub username. He will share the Docker container referenced by the docker-compose.yml file with you.
3. Follow the Getting Started Guide to set up your Docker Container Deployment
4. Interact with the Gateway Docker Container in the same way as you would with any other Gateway
5. Tell us about your experiences. You can either email Rod directly or post comments in the "API Gateway" User Forum

Documentation Notes

Please take these into account as you work through the documentation. We will address these very soon.

1. "Add Certificate for Local Registry" Section:

1. There is not need to Verify that the Docker Image can be pulled from a Local Registry for

Silent Install and configuration

- **The Problem:**

The Gateway can only be configured using an interactive text based menu. This makes it difficult to integrate the Gateway with configuration management and DevOps tools such as Chef, Puppet, and Ansible

- **Solution:**

Create a means to silently configure the Gateway which allows it to be configured over SSH.

Silent Install and configuration Functionality

- Create Gateway Container Cluster
- Join Existing Gateway Docker Cluster

developer



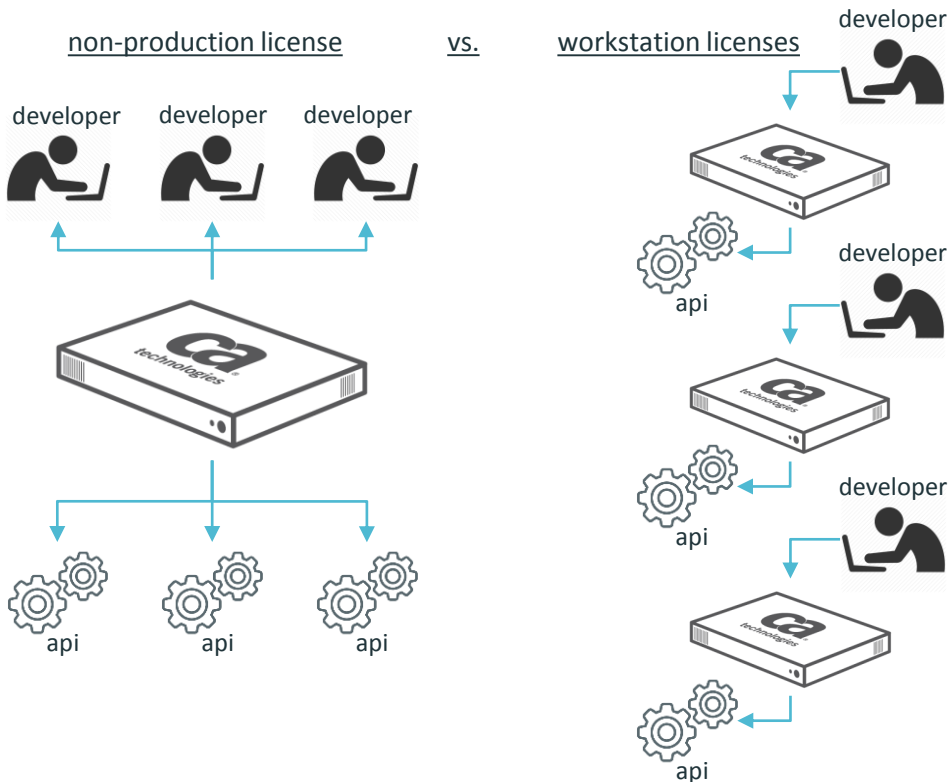
```
rmoshfeghi@rmosh-pc ~  
$ cat create-node.properties | ssh ssgconfig@gatewayhost /opt/SecureSpan/Gateway/config/bin/ssgconfig-headless create
```

```
create-node.properties  
1  ### Headless config create template properties file ###  
2  
3  ### Node Configuration ###  
4  ## Node Enabled State  
5  node.enable=true  
6  ## Configure the node.properties  
7  configure.node=true  
8  
9  
10 ### Cluster Configuration ###  
11 ## Cluster Hostname  
12 cluster.host=10.242.14.118  
13 ## Cluster Passphrase  
14 cluster.pass=7layer  
15  
16  
17 ### Database Connection ###  
18 ## Creates the database  
19 configure.db=true  
20 ## The database type, either 'mysql' or 'embedded'  
21 database.type=mysql  
22 ## Database Hostname  
23 database.host=localhost  
24 ## Database Port  
25 database.port=3306
```



Developer License for Gateway

Developer Workstation License



Able to provide individual use Gateways for development purposes

- Low cost for individual licenses, takes advantage of new deployment model.
- Easy to setup, deploy, manage. Allows individual developers to have individual development environments for Policy and API's

Running on new CentOS based virtual appliance

Publish Swagger based API's

Allow Gateway administrators, developers, and policy authors to publish API endpoints using Swagger documents.

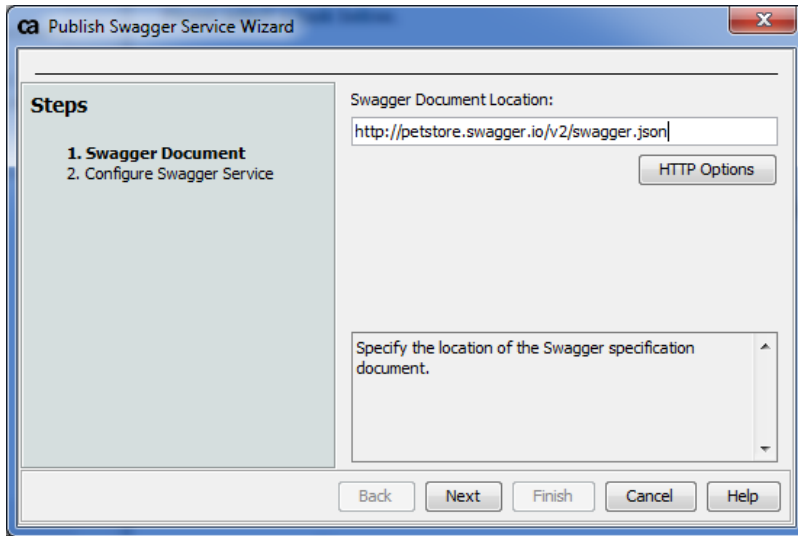
What is Swagger?

Popular representation syntax for RESTful API's and services

- Can be represented both in JSON or XML
- Declarative resource specification, no need for knowledge of server implementation or access to server code
- Allows developers to quickly interact with a remote service, within minimal implementation logic

```
    ],
    "paths": {
      "/pet": {
        "post": {
          "tags": [
            "pet"
          ],
          "summary": "Add a new pet to the store",
          "description": "",
          "operationId": "addPet",
          "consumes": [
            "application/json",
            "application/xml"
          ],
          "produces": [
            "application/xml",
            "application/json"
          ],
          "parameters": [
            {
              "in": "body",
              "name": "body",
              "description": "Pet object that needs t",
              "required": true,
              "schema": {
                "$ref": "#/definitions/Pet"
              }
            }
          ]
        }
      }
    }
  }
}
```

Publish Swagger based API's



Invoke publishing wizard via tasks menu

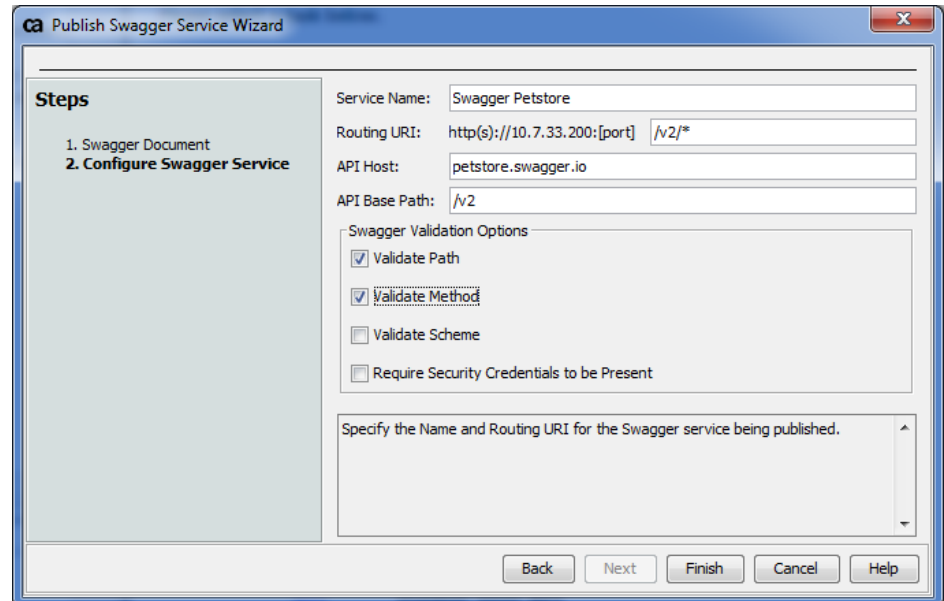
Enter location of Swagger document

- Used to define validation parameters in published service
- Used for routing
- Not permanently stored on Gateway with service

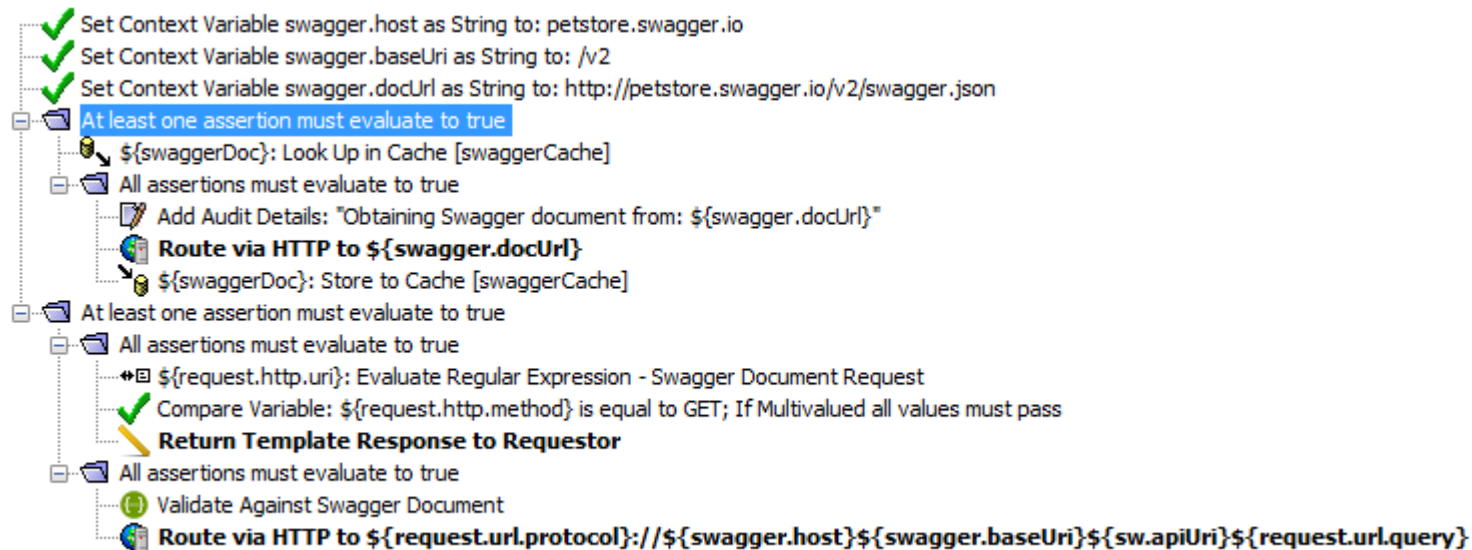
Enter information about the API for the Gateway to use

Optional validation options:

- Gateway can use information from the swagger document to validate/filter requests and responses, before they go to the backend service



Publish Swagger based API's



Invoke publishing wizard via tasks menu

Enter location of Swagger document

- Used to define validation parameters in published service
- Used for routing
- Not permanently stored on Gateway with service

ca Validate Against Swagger Document Pr...

Swagger Document: * ☒ OK

Service Base:

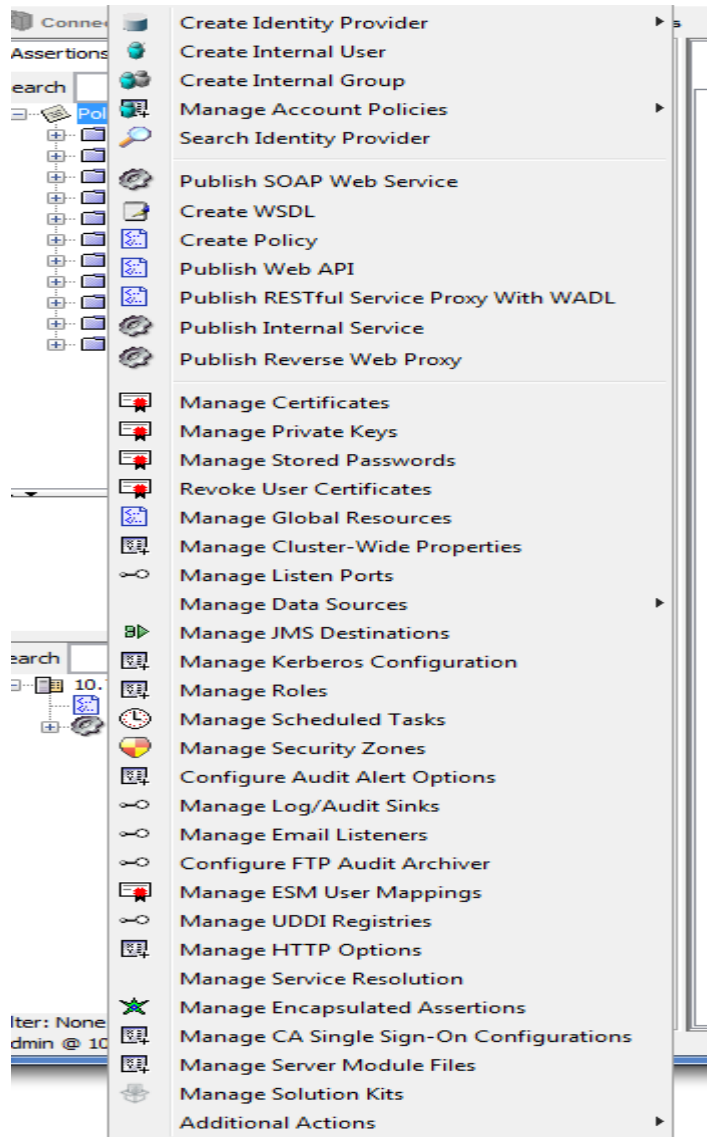
Validation Options

- ☒ Validate Path
- ☒ Validate Method
- ☐ Validate Scheme
- ☐ Require Security Credentials to be Present

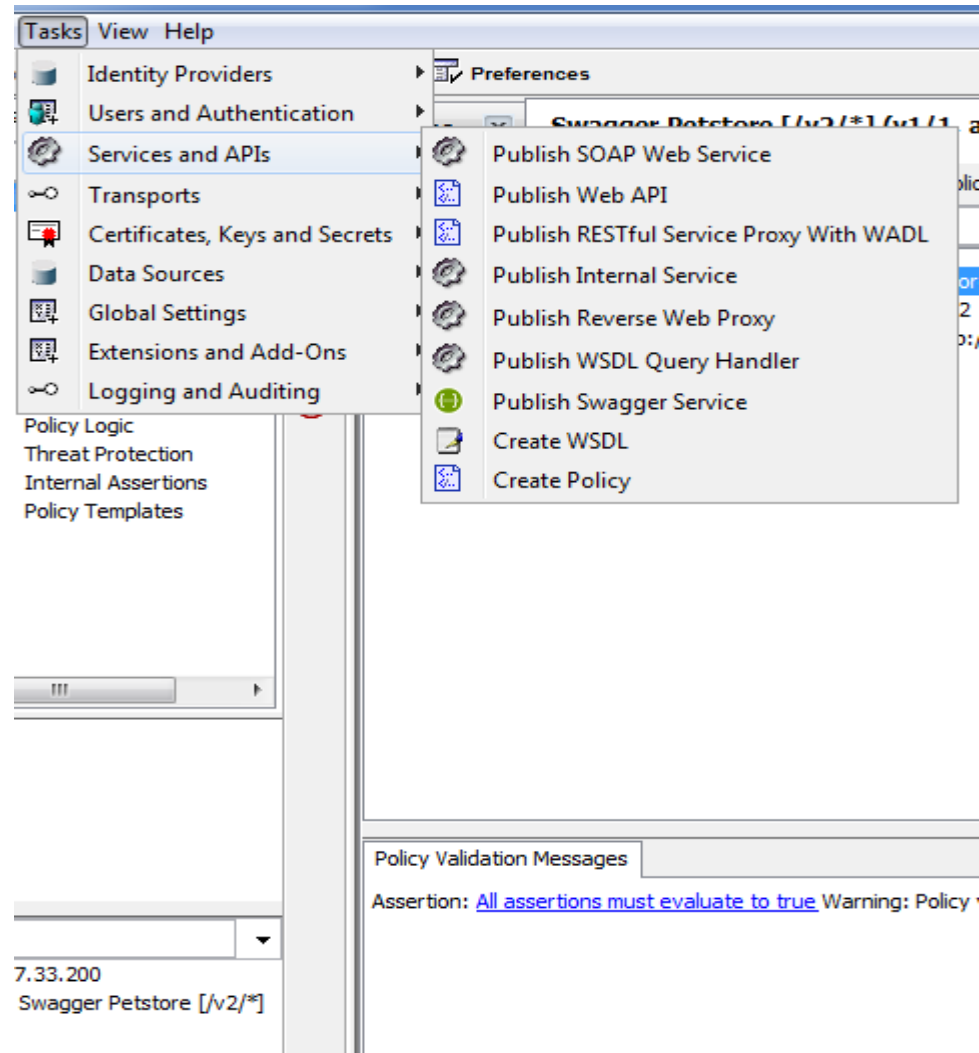
Variable Prefix: * ☒ OK

Policy Manager Task menu

9.0



9.1



Policy Manager speed improvements

The Problem:

Connecting the Policy Manager to geographically distant Gateways resulted in very slow login times, and slow refresh times when connected

Solution:

Changes to the way the Policy Manager loads information from the Gateway, and refreshes that information while a session is active.

Resulted in anywhere from 33% to 75% faster login times.



CA Single Sign On (Siteminder) request/response cache

CA Single Sign-On Configuration Properties

Configuration Name: * CA Single Sign On (production) Register

Agent Configuration

Secret: *

Address: 127.0.0.1 ☐ Check IP

Host Name: * prodman91.ca.com

FIPS Mode: * COMPAT

Cluster Threshold: * 50 ☐ Enable Failover ☒ Update SSO Token

Cluster Settings

Name	Value
authenticationCache.maxAge	360000
authenticationCache.size	1000
authorizationCache.maxAge	360000
authorizationCache.maxSize	1000
server.0.0.address	10.7.42.11

Add Edit Remove

☐ Disable Test OK Cancel

Added per-agent request and response caching functionality to the Gateway

- Allows users to configure caches for authentication/authorization requests and responses
- Sizes and lifespans of cache values are entirely user configurable, and can be optionally disabled
- When enabled, significant performance improvements for CA SSO authN/authZ via the Gateway
- Supports multiple agents, can configure cache properties for each agent separately.

Audit Subsystem Improvements

Configure the Gateway to bypass auditing subsystem when database is full

In previous versions, Gateway would stop processing traffic when the database filled up with audits

Introduce a new cluster property in Musket:

audit.managementStrategy	STOP/BYPASS
---------------------------------	--------------------

STOP: default behaviour, Gateway will stop processing traffic until space is made available in database

BYPASS: Gateway will continue processing traffic, and will bypass the audit subsystem

Functionality gives users options on how to configure their Gateway to behave when the database is full, allows them to avoid unexpected data processing suspensions.

Other Improvements

- Replay protection in clustered environments
- Thales HSM – compatibility with JDK8
- Support for RHEL 7 in software form factor
- Increased support for cloud-init in Gateway AMI form factor



Aaron Flint


Senior Product Manager
aaron.flint@ca.com

Hirbod (Rod) Moshfeghi

Senior Product Manager
hirbod.moshfeghi@ca.com

 @CA_Community

 slideshare.net/CAinc

 linkedin.com/company/ca-technologies

communities.ca.com