Federator.ai®

- Up to 70% resource savings
- Increased operational efficiency
- Reduced manual configuration time

Challenges

Container adoption is growing, and Kubernetes is becoming the de facto standard of container management platforms. Whether container adoption occurs on-premises, in public clouds, or both, the operational overhead is enormous. IT administrators cannot foresee computing resource demands of applications, so they must reserve more computing resources for a workload than needed. Managing computing resources and optimizing costs on multiple clouds are daunting tasks. Federator.ai, ProphetStor’s Artificial Intelligence for IT Operations (AIOps) platform, provides intelligence to orchestrate container resources on top of VMs (virtual machines) or bare metal, allowing users to operate applications without the need to manage the underlying computing resources.

Federator.ai Overview

Over-provisioned computing resources and the deployment of the incorrect number and/or size of VMs and/or pods are two common issues in a cloud-native environment. Federator.ai addresses these problems by orchestrating resources in multi-cloud environments. As shown in Figure, Federator.ai optimizes costs for both Day-1 deployment and Day-2 operations. It utilizes metrics from monitoring systems, Datadog, Sysdig, or Prometheus, to predict resource consumption dynamically and recommends the right amount of resources for pods, providing a 20 – 70% reduction of wasted resources for a typical workload, as well as preventing under-provisioning of resources for mission-critical workloads. Users can stack up the predicted pod resources to determine the right number and size of VMs to deploy and enable the automatic execution of these recommendations.

With Federator.ai, users no longer need to specify the CPU and memory requests and limits for each container. It recommends optimal pod configurations. The direct effect is that the configured resources will accurately and dynamically match the workload. It also effectively reduces occurrences of under-provisioned issues, such as out-of-memory (OOM).
Key Features
After Federator.ai is deployed in any Kubernetes or OpenShift environment, it learns application resource usage patterns and predicts the needed resources down to the container level. Federator.ai also provides dashboards that display the recommendations for different layers of resources in a Kubernetes cluster or a VM cluster.

Multi-layer workload prediction
Federator.ai applies multiple analytics tools, such as machine learning and signal processing, to predict containerized application, namespace, and node resource usage in multiple clusters as the basis for resource recommendations. Federator.ai supports both physical and virtual CPUs and memories.

Application-aware recommendation execution
The application resource demand determines the number and size of containers. Federator.ai utilizes resource usage prediction based on workload patterns to recommend the right sizes and/or right number of containers and automatically scale the containers to handle the demands.

Supported Metrics Data Sources
Federator.ai supports metrics collected through Prometheus, Datadog, or Sysdig monitoring services for Kubernetes clusters and VMware/vCenter, AWS CloudWatch for VM clusters.

Policy-driven planning of CPU and memory
Federator.ai plans cluster-wide CPU and memory allocation for different types of applications according to the policy specified by users.

Enterprise-ready
Federator.ai is designed to work with any Kubernetes and OpenShift-operated environment. Federator.ai provides application lifecycle management based on the Operator Framework and works seamlessly with Kubernetes and Red Hat OpenShift.

Easy installation
Installing Federator.ai is easy as it works as an Operator on Kubernetes and OpenShift. In addition, Federator.ai can also be installed through Ansible playbook or Helm charts.

Continuous recommendations for optimal resource planning
Federator.ai continuously generates recommendations and learns better with more accumulated metrics data.

Benefits
Federator.ai aims to provide optimal resource planning recommendations that will help enterprises make better decisions. The benefits of Federator.ai include:

Up to 70% resource savings
Federator.ai mainly serves to reduce unnecessary spending and increase application service quality for both enterprises and cloud providers. ProphetStor data scientists and engineering teams work together to build the most advanced AIOps solution to reduce resource wastage at different infrastructure layers. With the help of patented prediction technologies Federator.ai simultaneously reduces spending and delivers necessary performance.

Increased operational efficiency
Federator.ai frees users from continuously monitoring Kubernetes / OpenShift cluster or VM cluster utilization and cloud spending. Users also do not need to manually record usage data calculate optimal configurations, and change configurations based on the calculations. These tasks are routinely accomplished when using Federator.ai.

Reduced manual configuration time with digital intelligence
Federator.ai allows users to turn on the optimization engine any time. Using Federator.ai open API, users can re-configure pods with the right values at the right time.

Fully-integrated with leading cloud monitoring services
## Feature Details and Specifications

| AI-based multilayer workload predictions | - Workload predictions for multilayer Kubernetes resources: clusters, nodes, namespaces, applications, and controllers  
|                                           | - Workload predictions for individual VMs and VM clusters  
|                                           | - Generate daily, weekly, and monthly workload predictions  |
| Intelligent recommendations for resource planning | - Rightsizing CPU/memory resource recommendations for Kubernetes clusters, nodes, namespaces, applications, and controllers to reduce waste without compromising performance  
|                                           | - Extending resource usage predictions and recommendations for VM nodes and VM clusters  |
| Proactive and application aware autoscaling | - Intelligent autoscaling of containers based on workload predictions  
|                                           | - Autoscale containers based on application-specific workload metrics and KPI’s  
|                                           | - Autoscale Nginx ingress upstream services  
|                                           | - Autoscale Kafka consumers  |
| Auto provisioning application resources | - User-defined provisioning policy for automatic adjusting resources allocation for application container and namespaces  
|                                           | - Additional headroom adjustments and min/max resource allocation settings  
|                                           | - Flexible scheduling options  
|                                           | - Downloadable resource provisioning scripts for easy customization and 3rd party integration  |
| CI/CD integration | - Simple integration to CI/CD pipeline for right-sized containers deployment  
|                                           | - Sample script for easy Terraform deployment integration  |
| Multicloud cost analysis | - Recommend most cost-effective cluster configuration for AWS, Google, and Azure  
|                                           | - Recommendations based on on-demand, reserved, and Spot instances  
|                                           | - Workload prediction-based recommendations for individual VM instances  
|                                           | - Custom price book for cost calculation of on-prem deployments  
|                                           | - Automatic update of price books from public cloud service  |
| Application cost analysis | - Cost analysis for namespaces and applications based on resource usage  
|                                           | - Provide estimated cost savings based on recommended resource allocations  
|                                           | - Provide actual cost savings from auto provisioning policy  
|                                           | - Assist budget planning via cost forecasts for namespaces and applications  |
| VM cost analysis | - Cost analysis for each VM in a VM cluster based on resource usage and predictions  
|                                           | - Provide estimated cost savings based on recommended resource allocations  |
| Multiple metrics data sources | - Support metrics from Prometheus, Datadog, Sysdig, VMware/vCenter, AWS CloudWatch  |
### Installation
- Easy installation through Operator framework on Kubernetes/OpenShift
- Support Ansible Playbook installation
- Support installation by Helm charts

### Easy-to-use UI
- Visualization of resource usages and predictions for multilayer of Kubernetes resources
- Support monitoring for multiple clusters
- User defined application with controllers from multiple namespaces

### Integration with third party monitoring services
- Single pane of glass management from Datadog, Sysdig monitoring portal with custom dashboards
- Preconfigured Datadog Monitor for under-provisioned resource alerts

### Supported platforms
- Kubernetes v1.11x – v1.20x
- Red Hat OpenShift v3.11, v4.x
- Amazon EKS
- Google GKE
- MicroSoft AKS
- SUSE/Rancher v2.4.8, v2.5.8
- VMWare vCenter 5.5/6.0/6.5/6.7/7.0