AWS OpsWorks: Best Practices: Optimizing the Number of Application Servers

AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet. Chef and Puppet are automation platforms that allow you to use code to automate the configurations of your servers. OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed, and managed across your Amazon EC2 instances or on-premises compute environments. OpsWorks has three offerings, AWS Opsworks for Chef Automate, AWS OpsWorks for Puppet Enterprise, and AWS OpsWorks Stacks.

A production stack commonly includes multiple application servers distributed across multiple Availability Zones. However the number of incoming requests can vary substantially depending on time of day or day of the week. You could just run enough servers to handle the maximum anticipated load, but then much of the time you will end up paying for more server capacity than you need. To run your site efficiently, the recommended practice is to match the number of servers to the current request volume.

AWS OpsWorks Stacks provides three ways to manage the number of server instances.

- 24/7 instances are started manually and run until they are manually stopped.
- Time-based instances are automatically started and stopped by AWS OpsWorks Stacks on a user-specified schedule.
- Load-based instances are automatically started and stopped by AWS OpsWorks Stacks when they cross a threshold for a user-specified load metric such as CPU or memory utilization.

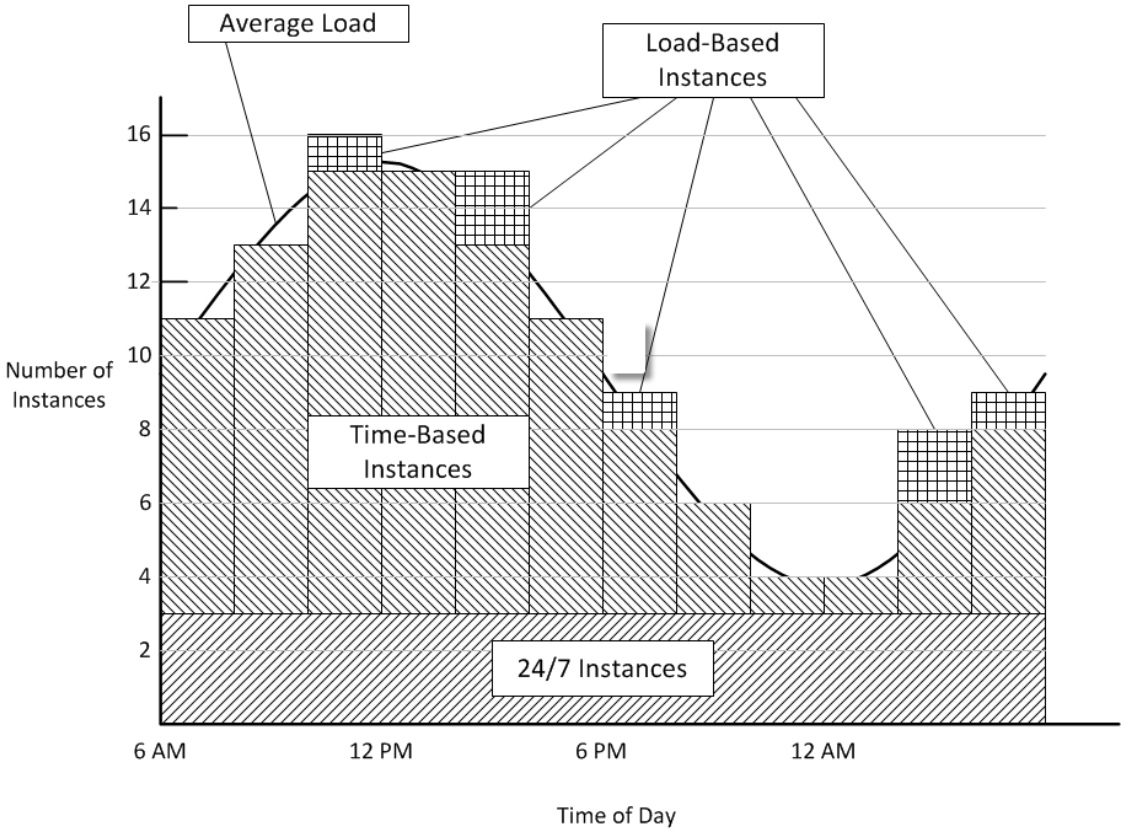
Note

After you have created and configured your stack's time and load-based instances, AWS OpsWorks Stacks automatically starts and stops them based on the specified configuration. You don't have to touch them again unless you decide to change the configuration or number of instances.

Recommendation: If you are managing stacks with more than a few application server instances, we recommend using a mix of all three instance types. The following is an example of how to manage a stack's server capacity to handle a variable daily request volume with the following characteristics.

- The average request volume varies sinusoidally over the day.
- The minimum average request volume requires five application server instances.
- The maximum average request volume requires sixteen application server instances.
- Spikes in request volume can usually be handled by one or two application server instances.

This is a convenient model for the purposes of discussion, but you can easily adapt it to any variation in request volume and also extend it to handle weekly variations. The following diagram shows how to use the three instance types to manage this request volume.



This example has the following characteristics:

- The stack has three 24/7 instances, which are always on and handle the base load.
- The stack has 12 time-based instances, which are configured to handle the average daily variation.
  One runs from 10 PM to 2 AM, two more run from 8 PM to 10 PM and 2 AM to 4 AM, and so on. For simplicity, the diagram modifies the number of time-based instances every two hours, but you can modify the number every hour if you want finer-grained control.
- The stack has enough load-based instances to handle traffic spikes that exceed what can be handled by the 24/7 and time-based instances.
  AWS OpsWorks Stacks starts load-based instances only when the load across all of the currently running servers exceeds the specified metrics. The cost for nonrunning instances is minimal (Amazon EBS-backed instances) or nothing (instance store-backed instances), so the recommended practice is to create enough of them to comfortably handle your maximum anticipated request volumes. For this example, the stack should have at least three load-based instances.

Note

Make sure you have all three instance types distributed across multiple Availability Zones to mitigate the impact of any service disruptions.

Source: https://docs.aws.amazon.com/opsworks/latest/userguide/best-practices-autoscale.html