2. Huge spikes of traffic

To accommodate huge spikes of traffic, several options are available for autoscaling. However, application behavior is also important, it is critical to know the effect of traffic/requests on different components of the provisioned EC2 instances which can be accomplished by monitoring default matrices along with the custom metrics such as RAM utilization, etc.
Once known, set up the CloudWatch alarm based on those metrics and trigger the autoscaling policy, scale-up EC2 instances if necessary.
Another way is to use a load balancer in front of the autoscaling group are pre-warming.
Yet, if the period of sudden traffic is known, one can also make use of scheduled scaling which could be cost-effective.

3. ElastiCache Redis as central PHP session storage – AutoScaling

Considering we have an autoscaling group that can launch new instances and terminates instances according to the setup. Below are the high-level steps to use a Redis cluster via ElastiCache as a central PHP session storage:

- Provision an ElastiCache Redis cluster, i.e, security group, Name, enable cluster mode, no of replicas, port node type, etc

- Get a configuration endpoint (master node address)

- Either create a new custom AMI of the application with tools required to access the Redis cluster such as redis-tools, Predis etc
  run a script to configure application (PHP session handler configuration) to use Redis endpoint using bootstrap
  or
- Use current AMI, install required tools and run a script to configure application (PHP session handler configuration) to use Redis endpoint using bootstrap or with the help of some configuration management tool such as Chef