xosphere

Instance Orchestrator User Guide

Table of Contents

Overview

Xosphere Instance Orchestrator is an intelligent orchestration engine that enables savings up to 80% on Amazon Elastic Compute Cloud (EC2) by optimizing the usage of Spot instances. Instance Orchestrator makes using Spot instances incredibly easy and also enables them to be used in many more use cases. Even with applications that have high availability requirements, Instance Orchestrator unlocks the savings potential of Spot instances in a turnkey fashion.

Instance Orchestrator is a cloud native application that installs into your Amazon Web Services (AWS) account. Installation is performed using either a CloudFormation stack or a Terraform module. For more details see the Installation with CloudFormation or Installation with Terraform sections.

Both stateless and stateful applications are supported by Instance Orchestrator. For stateless applications, Instance Orchestrator integrates with native AWS Auto Scaling groups. For stateful application, Instance Orchestrator uses collections of instances, called Xogroups, to manage the optimization process.

Instance Orchestrator uses an opt-in design. Individual instances or Auto Scaling groups need to be enabled in order for Instance Orchestrator to manage them. All configuration, including enabling management, is performed via AWS tags. Tags can be applied using any method or tool that is used within the organization to manage tags (for example, AWS Console, AWS CLI, AWS APIs, infrastructure-as-code platforms such as CloudFormation or Terraform, cloud management platforms, etc.).

Once the configuration tags have been applied, Instance Orchestrator will automatically perform its management duties on an on-going basis.

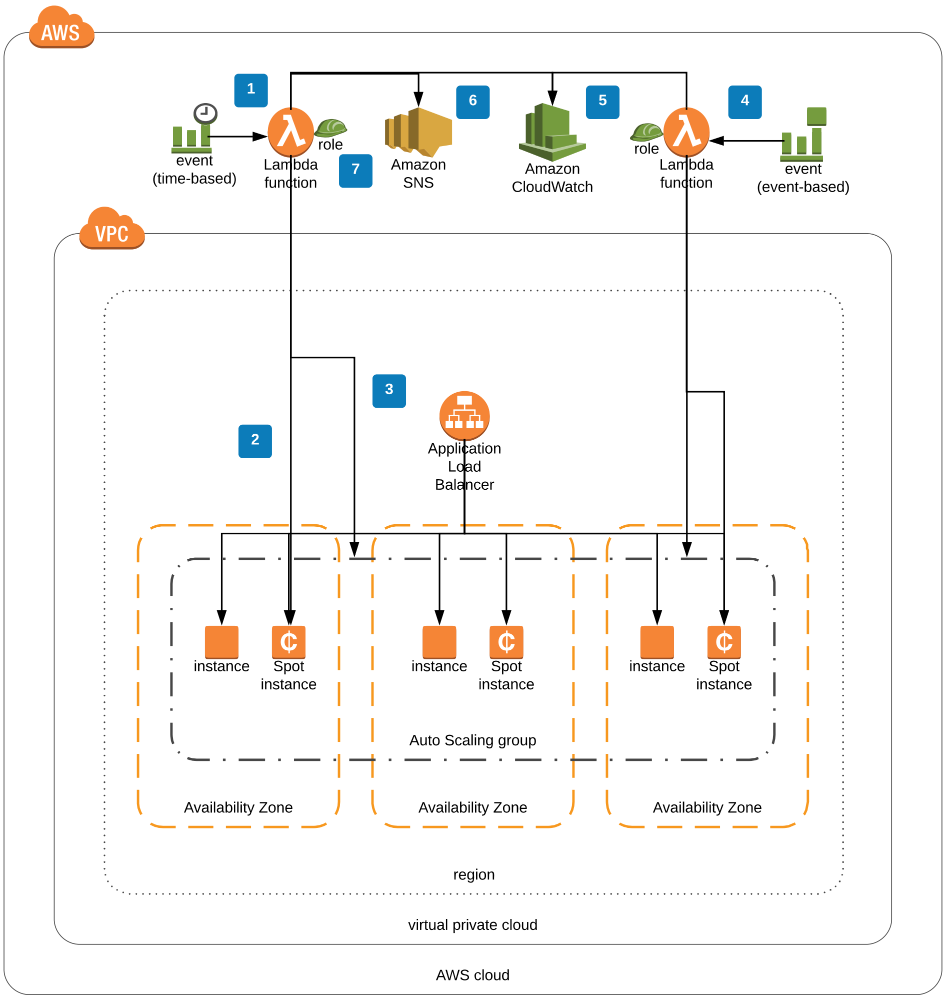
Architecture

Below is an architectural diagram that outlines the components of Instance Orchestrator and the corresponding AWS services that are leveraged:

Xosphere Instance Orchestrator

Running Xosphere on AWS

Xosphere Instance Orchestrator provides intelligent management of Spot instances giving applications the same level of reliability as On-Demand instances at a fraction of the cost. Xosphere orchestrates the replacement of On-Demand instances with Spot instances and vice-versa when the dynamics of the Spot market fluctuate. Using Xosphere can save up to 80% on EC2 costs without sacrificing reliability.



1 A time-based CloudWatch event triggers the core Instance Orchestrator Lambda function on a regular basis.

2 Instance Orchestrator looks for Xosphere enabled Auto Scaling Groups (via tags) and launches a Spot instance to replace an On-Demand instance.

3 Once the replacement Spot instance is fully bootstrapped and passing health checks, Instance Orchestrator attaches the instance to the Auto Scaling Group, detaches the On-Demand instance, drains any load balancer connections, and gracefully terminates the On-Demand instance.

4 When a Spot termination notification is received via a CloudWatch event, the Terminator Lambda function triggers a scale-out operation on the Auto Scaling Group, and begins to drain any connections on the terminating Spot instance.

5 Execution logs from Instance Orchestrator and Terminator are written to CloudWatch logs.

6 SNS notifications are optionally sent when instances are launched or terminated.

7 Each Lambda function leverages its own IAM role with least minimum privileges necessary to execute.

xosphere™

Planning and Guidance

Security

Xosphere Instance Orchestrator contains self-hosted subscription software that installs into your AWS account. Since Instance Orchestrator is not interacting with your account from a SaaS platform, there is no requirement to delegate any access using IAM roles. Xosphere will have no remote access to your account and all sensitive data within the account and generated by Instance Orchestrator will remain within the AWS account.

Xosphere Instance Orchestrator follows recommended best security practices and leverages a series of IAM roles and ensures each Lambda function executes with a role that has the least minimum privileges required to perform its duties. Note, that this means that CloudFormation stack or Terraform modules used to install Instance Orchestrator will require being run with a user account that has permissions to create IAM roles.

AWS Lambda runs your function code securely within a VPC by default. In most cases Instance Orchestrator does not interact directly with any underlying AWS resources, so no configuration is required to enable the Lambda functions to access resources inside your private VPC. Instead, Instance Orchestrator will leverage the permissions from its IAM role and interact with the AWS API to manage AWS resources within the account. The one exception to this is integration with Kubernetes to enable Pods and containers to gracefully drain when performing a replacement. For more details on enabling VPC access to your Kubernetes or EKS cluster see the Container Orchestration Integration section.

Instance Sizing

Xosphere Instance Orchestrator is a cloud native application and runs exclusively as AWS Lambda functions. Since there is no AMI and no EC2 instance to launch, there is no need to perform any instance sizing planning.

Costing

Since Xosphere Instance Orchestrator runs exclusively in AWS Lambda there is very minimal cost associated with running. If the AWS account is eligible for the Lambda Free Tier, Instance Orchestrator can run for free well within the limits of the Lambda Free Tier. If the AWS account is not eligible for the Lambda Free Tier, the total annual cost associated with running Instance Orchestrator is less than $20.

Account Creation

In order to use Xosphere services an Xosphere account must be provisioned. To sign up for a new account, please visit https://portal.xosphere.io/signup.

Multiple AWS accounts can be combined into a single Xosphere account for billing and reporting purposes. When completing this signup wizard, simply specify the AWS accounts that will be managed by Instance Orchestrator. You can always update and modify the list of linked AWS accounts by visiting the Xosphere Dashboard Settings.

After completing the signup wizard, Xosphere will generate a custom CloudFormation template for the organization with the required Xosphere API key embedded in it. To deploy this CloudFormation template please refer to the Installation with CloudFormation section.

Xosphere will provide the Launch Stack link for the custom CloudFormation template after the signup wizard is completed. The Launch Stack link will also be emailed to the account owner. The Launch Stack link can also be accessed at any time by visiting the Xosphere Dashboard.

Alternatively, Xosphere provides a Terraform module that can be used to install Instance Orchestrator. For more details please refer to the Installation with Terraform section.

To access the Xosphere Dashboard please visit https://dashboard.xosphere.io/.

Installation

In order for Xosphere Instance Orchestrator to perform its orchestration duties, Instance Orchestrator must be installed into each AWS account that is to be managed.

Installation with CloudFormation

To install Xosphere Instance Orchestrator in an AWS account, simply click on the CloudFormation Launch Stack link provided by the signup wizard or the Xosphere Dashboard (in the Installation section) to launch the CloudFormation stack.

After clicking the link, you will be directed to the Create Stack wizard in the CloudFormation console for the account you are logged into. If you are not currently logged into any AWS account, you will be presented with the AWS login page. After entering your credentials, you will be taken to the Create Stack wizard in the CloudFormation console.

On the Select Template step leave the default values and click the Next button.

On the Specify Details step, a list of parameters to the template will be presented:

> **AMICleanerScheduleExpression** – this parameter is used to set the Cron expression for the AMI Cleaner scheduled Lambda function. It is not common to change this parameter.
>
> **APITokenARN** – this parameter is used to specify the API Token Secret ARN stored in Secrets Manager in the Xosphere AWS account. You should not modify this parameter.
>
> **BudgetScheduleExpression** – this parameter is used to set the Cron expression for the Budget scheduled Lambda function. It is not common to change this parameter.
>
> **CodeDeployPassRoleARNResourcePattern** – this parameter is used to set the resource pattern used in IAM policies for granting privileges to PassRole for CodeDeploy deployments. By default, this value is set to * indicating Instance Orchestrator can PassRole to all CodeDeploy deployments. This is convenient for out of the box functionality. For a higher level of security, the parameter can be modified to specify an AWS resource pattern (with optional wildcards) to be used for the resource pattern in the IAM role policies.
>
> **CreateLoggingBuckets** – this parameter is used to specify where S3 access logging buckets are created for any Xosphere S3 data buckets that are created. By default, this value is set to false meaning no access logging buckets will be created. Set this value to true to enable S3 access logging buckets.
>
> **CustomerId** – this parameter is used to specify the Xosphere Customer ID. You should not modify this parameter.

DailyBudgetGracePeriod – this parameter defines the grace period (in seconds) for the Budget Enforcer for daily budgets. By default, this value is 1200 seconds (or 20 minutes) which indicates if the daily budget is exceeded within the last 20 minutes of the day, the Budget Enforcer will not terminate the instance and allow it to continue to run.

EnableAutoSupport – this parameter is used to enable publishing the Xospehre Auto Support functionality. By default, this value is set to true, meaning Auto Support is enabled. Auto Support will proactively identify common issues and notify you on how to resolve them, as well as provide Xosphere Support with access to the Instance Orchestrator logs whenever a support ticket is opened. While it is highly recommended to enable Auto Support, you can disable it by setting this parameter to false.

EnableCloudWatch – this parameter is used to enable publishing of CloudWatch metrics for license and usage information. This value is defaulted to false. Changing this value to true will result in additional charges for the CloudWatch metrics. Most installations will not need to change this parameter.

EnhancedSecurityManagedResources – this parameter is used to enable Enhanced Security mode. By default, this parameter is set to false meaning Enhanced Security mode is not enabled. Xosphere always leverages least minimum privileges required for IAM role policies, however some resource patterns may leverage wildcard patterns for the sake of management convenience. For organizations with strict security policies, it may be required to disable this wildcard access. Setting this parameter to true will enable Enhanced Security mode which will require administrators to add authorized tags to all AWS resources (such as AMIs, EBS volumes, etc.) that Instance Orchestrator may need access to in order to replace instances. Due to the added management overhead, it is not recommended to enable Enhanced Security mode unless your organization’s security policy mandates it.

EnhancedSecurityTagRestrictions – this parameter is used to enable Enhanced Security mode. By default, this parameter is set to false meaning Enhanced Security mode is not enabled. Xosphere always leverages least minimum privileges required for IAM role policies, however some resource patterns may leverage wildcard patterns for the sake of management convenience. For organizations with strict security policies, it may be required to disable this wildcard access. Setting this parameter to true will enable Enhanced Security mode which will require administrators to add authorized tags to all AWS resources (such as AMIs, EBS volumes, etc.) that Instance Orchestrator may need access to in order to replace instances. Due to the added management overhead, it is not recommended to enable Enhanced Security mode unless your organization’s security policy mandates it.

EnhancedSecurityUseCMK – this parameter is used to enable the use of KMS CMKs rather than the standard AWS service managed encryption keys. By default, this parameter is set to false meaning the standard AWS service managed encryption keys are used. Setting this parameter to true will generate CMKs to be used for encryption purposes. Note that use of CMKs will result in increased costs from AWS.

EnableECS – this parameter is used to ECS integration. The default value is true, meaning ECS integration is enabled. You csan disabled ECS integration by setting this parameter to false.

GroupInspectorScheduleExpress– this parameter is used to set the Cron expression for the Group Inspector scheduled Lambda function. It is not common to change this parameter.

K8sDrainTimeoutInMins – this parameter sets the timeout in minutes when draining Kubernetes nodes. The default value is 15 minutes, indicating that if Kubernetes takes longer than 15 months to drain a worker node then Instance Orchestrator will terminate the instance without waiting for the drain process to complete.

K8sVPCSecurityIds – this parameter is used to specify the security group IDs to use for Lambda function that communicates with the k8s cluster. This value defaults to none, meaning you are not using the k8s integration. Changing this value to will result in the Lambda function being created using the specified security group so that it can properly contact the k8s cluster. If you have more than one k8s cluster that reside in different subnets, you may use multiple security groups in a comma delimited list. The security groups ID(s) used for this parameter should be the same as the security group(s) used for the k8s worker nodes.

K8sVPCSubnetIds – this parameter is used to specify the subnet IDs to use for Lambda function that communicates with the k8s cluster. This value defaults to none, meaning you are not using the k8s integration. Changing this value to will result in VPC endpoints being created for the Lambda function in each subnet so that it can properly contact the k8s cluster. If you have more than one subnet, specify as a comma delimited list of subnet IDs. If you have more than one k8s cluster, specify a comma delimited list of all subnets across all clusters.

MonthlyBudgetGracePeriod – this parameter defines the grace period (in seconds) for the Budget Enforcer for monthly budgets. By default, this value is 36000 seconds (or 10 hours) which indicates if the monthly budget is exceeded within the last 10 hours of the month, the Budget Enforcer will not terminate the instance and allow it to continue to run.

PassRoleARNResourcePattern – this parameter is used to set the resource pattern used in IAM policies for granting privileges to PassRole. By default, this value is set to * indicating Instance Orchestrator can PassRole to any role that has a trust relationship with the EC2 service. This is convenient for out of the box functionality. For a higher level of security, the parameter can be modified to specify an AWS resource pattern (with optional wildcards) to be used for the resource pattern in the IAM role policies.

Regions – this parameter is used to provide the list of regions that Instance Orchestrator will support. This parameter accepts a comma delimited list of AWS regions. By default, this value is set to us-east-1,us-west-2. It is common to change this parameter to use the list of regions being actively used in the AWS account.

ScheduleExpression – this parameter is used to set the Cron expression for the Instance Orchestrator scheduled Lambda function. The default value is a Cron expression indicating the function will be run every two minutes. It is not common to change this parameter.

SnapshotCreatorScheduleExpression – this parameter is used to set the Cron expression for the Snapshot Creator scheduled Lambda function. The default value is a Cron expression indicating the function will be run every 15 minutes. It is not common to change this parameter.

SNSARNResourcePattern – this parameter is used to set the resource pattern used in IAM policies for granting privileges to publish SNS notifications. By default, this value is set to * indicating Instance Orchestrator can publish notifications to all SNS topics. This is convenient for out of the box functionality. For a higher level of security, the parameter can be modified to specify an AWS resource pattern (with optional wildcards) to be used for the resource pattern in the IAM role policies.

TerraformAWSProviderVersion – this parameter is used to set the version number of the Terraform AWS Provider to be used with the Xogroup integration with Terraform which updates the instance ID stored in the Terraform state file when an instance is replaced.

TerraformBackendAWSRegion – this parameter is used to set the AWS region where the Terraform backend S3 is hosted.

TerraformBackendDynamoDBTable – this parameter is used to set the DynamoDB table used for Terraform state file locking.

TerraformBackendS3Bucket – this parameter is used to set the S3 bucket used to store the Terraform state file.

TerraformBackendS3Key – this parameter is used to set the S3 key used to store the Terraform state file.

TerraformVersion – this parameter is used to set the version number of the Terraform binary to be used with the Xogroup integration with Terraform which updates the instance ID stored in the Terraform state file when an instance is replaced.

UpgradeScheduleExpression – this parameter is used to set the Cron expression for the Instance Orchestrator upgrade check scheduled Lambda function. The default value is a

Cron expression indicating the function will be run once a day. It is not common to change this parameter.

Version – this parameter is used by the CloudFormation stack to set version information. Note that this field cannot be changed to any other value.

After making any changes to the default values for the parameters, click the Next button.

On the Options page, optionally set any options that may be required for the target AWS account. In most cases, you can safely leave the default options on this step. Click the Next button to continue.

On the Review page, you must check the box that reads “I acknowledge that AWS CloudFormation might create IAM resources with custom names” before clicking on the Create button to launch to stack.

Installation with Terraform

Xosphere Instance Orchestrator can be installed using Terraform. Xosphere has published a native Terraform module in the Terraform Module Registry. To include the Xosphere Terraform module in a Terraform configuration simply add the following snippet to the Terraform configuration file (with the .tf extension).

```
provider "aws" {
  alias  = "us-west-2"
  region = "us-west-2"
}

provider "aws" {
  alias  = "us-east-1"
  region = "us-east-1"
}

module "xosphere_instance_orchestrator" {
  source          = "xosphere/instance-orchestrator/xosphere"
  customer_id     = "XXXXXXXXXXXXXXXXXXXXXXX"
  regions_enabled = ["us-east-1", "us-west-2"]
  providers = {
    aws = aws.us-west-2
  }
}

module "xosphere_event_relay_us_west_2" {
  source                   = "xosphere/instance-orchestrator-event-relay/xosphere"
  event_relay_iam_role_arn = module.xosphere_instance_orchestrator.event_relay_iam_role_arn
  event_router_sqs_url     = module.xosphere_instance_orchestrator.event_router_sqs_url
  installed_region         = module.xosphere_instance_orchestrator.installed_region
  xosphere_version         = module.xosphere_instance_orchestrator.xosphere_version
  providers = {
    aws = aws.us-west-2
  }
}

module "xosphere_event_relay_us_east_1" {
  source                   = "xosphere/instance-orchestrator-event-relay/xosphere"
  event_relay_iam_role_arn = module.xosphere_instance_orchestrator.event_relay_iam_role_arn
  event_router_sqs_url     = module.xosphere_instance_orchestrator.event_router_sqs_url
  installed_region         = module.xosphere_instance_orchestrator.installed_region
  xosphere_version         = module.xosphere_instance_orchestrator.xosphere_version
  providers = {
    aws = aws.us-east-1
  }
}
```

The customer_id value should be replaced with the actual Xosphere customer ID for your Xosphere account. This snippet can be automatically generated, including your customer ID, by navigating to the Installation section of the Xosphere Dashboard. Under the Terraform heading, simply select the desired regions to install and click on the Show Script button.

Configuration

Enabling Auto Scaling Groups

For stateless applications, Instance Orchestrator only executes on Auto Scaling groups that have explicitly been enabled. You can enable an Auto Scaling group by applying an AWS tag to the Auto Scaling group. The tag can be applied by any method used to manage AWS tags including: AWS Console, AWS CLI, AWS APIs, infrastructure-as-code (such as CloudFormation or Terraform), cloud management platforms, etc. The name and value of the of the tag must be:

xosphere:instance-orchestrator:enabled = true

Enabling Xogroups

For stateful applications, an Xogroup is used to define a collection or group of instances. Instance Orchestrator only executes on Xogroups that have explicitly been configured. You can create an Xogroup by applying an AWS tag to each instance in the group. The tag can be applied by any method used to manage AWS tags including: AWS Console, AWS CLI, AWS APIs, infrastructure-as-code (such as CloudFormation or Terraform), cloud management platforms, etc. The name and value of the of the tag must be:

xosphere:instance-orchestrator:xogroup-name = <name of group>

The name of the group can be any text string. Each instance in the Xogroup must have the same group name tag value.

Optional Configuration Settings

Additional tags can be applied to Auto Scaling groups or Xogroups to control various settings for Instance Orchestrator. In the case of Auto Scaling groups, these tags should be applied directly to the Auto Scaling group. In the case of Xogroups, these tags can be applied to any instance within the Xogroup. Only one instance in the Xogroup needs to be tagged with the optional configuration tags and that setting will be applied to all instances in the Xogroup. Alternatively, Xogroup options can also be managed by using the Xosphere Dashboard in the Xogroups section.

The optional configuration setting available include:

Account for Nitro

xosphere:instance-orchestrator:account-for-nitro

When determining compatible instance types for replacement, should the Nitro System be considered. By default this value is false, meaning the Nitro System compatability will not be considered.

Alert Topic ARN

xosphere:instance-orchestrator:alert-topic-arn

For Xogroups, the SNS topic ARN to send notification if an Xogroup alert is generated.

xosphere:instance-orchestrator:allowed-instance-types

Explicit list of allowed instances types for this Auto Scaling group. Accepts a comma-delimited list of instance types (e.g. m4.large,m5.large,c5.large,r4large). By default, this value is not set, meaning Instance Orchestrator will automatically determine eligible instance types based on the specs of the On-Demand instance being replaced.

Always On-Demand

xosphere:instance-orchestrator:always-on-demand

Flag indicating whether or not On-Demand instances are eligible to be replaced with Spot instances. Accepts Boolean values true or false. By default, this value is false, meaning the On-Demand instance is eligible to be replaced by a Spot instance. Note: this tag should be applied directly to an individual instance and should not be applied to an Auto Scaling group.

Authorized

xosphere:instance-orchestrator:authorized

When using Enhanced Security mode, this tag is to be applied to resources such as AMIs, EBS volumes, ENIs, or any other resource, in order to provide Instance Orchestrator with the IAM privileges required to access the resource while launching replacement instances.

Bid Multiplier

xosphere:instance-orchestrator:bid-multiplier

The bid multiplier when bidding on Spot instances. Accepts floating point values great than 0.0. By default, this value is 1.0, meaning Instance Orchestrator will place Spot bids for 100% of the On-Demand instance price.

Budget Name

xosphere:instance-orchestrator:budget-name

The name of the Budget defined in the Xosphere Dashboard to be used for this Auto Scaling group, Xogroup, or individual instance. By default, this value is null indicating that the Budget functionality is not enabled for this Auto-Scaling group, Xogroup, or individual instance. To enable the Budget functionality, set this value to match the name of a Budget defined in the Xosphere Dashboard. Note that budget names are case sensitive. Budgets can be used in conjunction with Spot instance orchestration in Auto-Scaling groups or Xogroups, or it can be used standalone on an individual EC2 instances that are not leveraging Spot instance orchestration. When Budgets are enabled, Instance Orchestrator will collect cost information and record budget usage. Optionally budgets can be configured to notify administrators when the budget usage has exceeded a specified threshold. Instance Orchestrator can also optionally enforce budgets that have been exceeded in which case instances will be stopped until the next budget period (month or day) begins.

Enable Burstable

xosphere:instance-orchestrator:enable-burstable

Allow On-Demand instances to be replaced by equivalent burstable Spot instances (e.g. the t2 family). Accepts true or false. By default, this value is false, meaning Instance Orchestrator will not replace any non-burstable On- Demand instance with a burstable Spot instance.

Enable CodeDeploy

xosphere:instance-orchestrator:enable-codedeploy

Flag indicating whether CodeDeploy integration is enabled. By default, this value is set to true, meaning the appropriate CodeDeploy deployment will be triggered during instance replacement. Setting this value to false will disable the CodeDeploy deployment during replacement. Note this setting is only applicable for instances associated with CodeDeploy. If an instance is not associated with CodeDeploy, it is not necessary to disable this setting.

Enable CloudWatch

xosphere:instance-orchestrator:enable-cloudwatch

Flag indicating whether or not CloudWatch metrics should be published. Accepts Boolean values true or false. By default, this value is set to false, meaning no CloudWatch metrics will be published. Note, setting this value to true will incur additional AWS expense for custom CloudWatch metrics.

Excluded Instance Types

xosphere:instance-orchestrator:excluded-instance-types

Explicit list of disallowed instance types or families for this Auto Scaling group. Accepts a comma-delimited list of instance types (e.g. m4.large,m5.large,c5.large,r4large), instance families (e.g. m4,m5,c5,r4), or a combination of both instance types and families (e.g. m4.large,c5,r5). By default, this value is not set, meaning Instance Orchestrator will automatically determine eligible instance types based on the specs of the On-Demand instance being replaced.

Ignore Load Balancer Health Check

xosphere:instance-orchestrator:ignore-health-check

Flag indicating whether to honor any applicable load balancer health check when performing instance replacements. By default, this value is set to true, meaning the load balancer health check will be enforced and instance replacements will only continue after the health check returns a healthy status. Setting this value to false will disable the load balancer health check during instance replacement and the instance replacement will continue regardless of the health check status.

Kubernetes Drain Timeout

xosphere:instance-orchestrator:k8s-drain-timeout-in-mins

Number of minutes to wait for Kubernetes pods to drain during an instance replacement. Accepts integer values. By default, this value is set to 15, meaning Instance Orchestrator will wait up to 15 minutes when draining Kubernetes pods during an instance replacement. Note that if the instance being replaced is a Spot instance reverting to an On-Demand instance, then the two-minute Spot termination notice period will prevail regardless of the drain timeout specified.

Kubernetes Pod Eviction Grace Period

xosphere:instance-orchestrator:k8s-pod-eviction-grace-period

The grace period in seconds when evicting pods during draining. The default value is -1 which means use the value specified by the pod (which is 30 seconds unless overridden).

Launch Instance SNS Topic

xosphere:instance-orchestrator:instance-launch-topic-arn

The ARN of an SNS topic to publish a message whenever an instance is launched. By default, this value is set to null indicating Instance Orchestrator will not publish any SNS messages when a new instance is launched.

Launch Instance Error SNS Topic

xosphere:instance-orchestrator:instance-launch-error-topic-arn

The ARN of an SNS topic to publish a message whenever an instance launch fails. By default, this value is set to null indicating Instance Orchestrator will not publish any SNS messages when a new instance launch fails.

Maximum Spot Pool Percent

xosphere:instance-orchestrator:max-spot-pool-percent

Maximum percentage of instances allowed to be running in a single Spot pool. Accepts floating point values from 0.0 to 100.0. By default, this value is set to 25.0, meaning up to 25% of instances in the Auto Scaling group or Xogroup can be placed in a single Spot pool.

Minimum On-Demand

xosphere:instance-orchestrator:min-on-demand

Minimum number of On-Demand instances to be running in this Auto Scaling group. Accepts integer values from 0 to the total number of instances in this Auto Scaling group. By default, this value is set to 0, meaning all On- Demand instances in the Auto Scaling group can be replaced with Spot instances.

Notify on Spot Launch

xosphere:instance-orchestrator:notify-on-spot-launch

Flag indicating whether SNS notifications should be sent when Spot instances are launched in this Auto Scaling group. Accepts Boolean values true or false. By default, this value is set to false, meaning no messages are sent when Instance Orchestrator launches a replacement Spot instance.

Optimize Reserved Instances

xosphere:instance-orchestrator:optimize-reserved-instances

Boolean value indicating whether to consider replacing On-Demand instances with On-demand instances of another instance class in order to take better advantage of under-utilized reserved instances. By default, this value is set to true, indicating that under-utilized reservations of compatible instance classes will be considered first before launching Spot instances.

Parallel Processing

xosphere:instance-orchestrator:parallel-processing

The number or percentage of instances to replace at the same time within a single Auto Scaling group or Xogroup. Numeric values with no percent sign will be treated as the number of instances, whereas numeric values followed by a percent sign (e.g. 10%) will be treated as a percentage of the number of instances in the group. The maximum simultaneous replacements made within a single group is 20 instances. If this value is set to a higher numeric value, or if the percentage calculation yields a number of instances greater than 20, only 20 replacements will happen in parallel. By default, this value is 1, meaning one instance will be replaced at a time.

Percent On-Demand

xosphere:instance-orchestrator:percent-on-demand

Percentage of On-Demand instances to be running in this Auto Scaling group. Accepts floating point values from 0.0 to 100.0. By default, this value is set to 0.0, meaning all On-Demand instances in the Auto Scaling group can be replaced with Spot instances.

Prefer Reserved Instances

xosphere:instance-orchestrator:prefer-reserved-instances

Flag indicating whether or not unused Reserved instances should be taken into account when deciding whether or not to replace an instance with a Spot instance. Accepts Boolean values true or false. By default, this value is set to true, meaning an On-Demand instance that is currently using a Reserved reservation will not be replaced by a Spot instance. This setting can be useful in scenarios where Reserved instances have been purchased or can be used in another account that is linked to the current account for billing purposes.

Rebalance Recommendations

xosphere:instance-orchestrator:rebalance-recommendation-enabled

Flag used to indicate whether a Rebalance Recommendation should trigger a replacement of a Spot instance back to an On-Demand instance. By default, this value is set to false, meaning instances are not replaced when a Rebalance Recommendation is received. Setting this value to true will trigger replacements whenever a Rebalance Recommendation event is received. This can be useful if you desire a more proactive replacement when there is volatility in the Spot pool that may cause a Spot termination to happen in the near future.

Revert Spot Instance

xosphere:instance-orchestrator:revert

Flag used to trigger the replacement of a Spot instance back to an On-Demand instance. Setting this value to true will trigger a replacement to happen. This setting is useful to simulate a Spot instance termination event for testing purposes. Note this tag will automatically be deleted once the revert process occurs. Note that under normal circumstances, after the On-Demand instance replacement is complete, the new On-Demand instance will subsequently be replaced with a Spot instance. To avoid this, you can set the xogroup-enabled tag to false before setting the revert tag to true.

Schedule Enabled

xosphere:instance-orchestrator:schedule-enabled

Flag indicating whether the schedule is enabled or not. By default, this value is set to true indicating the schedule is enabled and instanced will be stopped and started at the appropriate time based on the schedule definition. Setting this value to false will disable the schedule and instances will not be stopped or started.

Schedule Mode

xosphere:instance-orchestrator:schedule-mode

Schedules have two modes: strict or permissive. Strict mode indicates that if the instance state is changed outside of Instance Orchestrator (i.e. the instance is stopped or started), then

Instance Orchestrator will enforce the schedule and override the manual change on the next cycle. In permissive mode, changes to the instance state (i.e. if the instance is stopped or started) made outside of Instance Orchestrator will be allowed and the schedule will not override the instance state until the next scheduled state transition. By default, this value is set to permissive mode.

Schedule Name

xosphere:instance-orchestrator:schedule-name

The name of the Schedule defined in the Xosphere Dashboard to be used for this Auto Scaling group, Xogroup, or individual instance. By default, this value is null indicating that the Schedule functionality is not enabled for this Auto-Scaling group, Xogroup, or individual instance. To enable the Schedule functionality, set this value to match the name of a Schedule defined in the Xosphere Dashboard. Note that schedule names are case sensitive. Schedules can be used in conjunction with Spot instance orchestration in Auto-Scaling groups or Xogroups, or it can be used standalone on an individual EC2 instances that are not leveraging Spot instance orchestration. When Schedules are enabled, Instance Orchestrator will automatically stop and start instances according to the defined schedule.

Suspend ASG Availability Zone Rebalance

xosphere:instance-orchestrator:suspend-asg-az-balance

Flag indicating whether the Auto Scaling group AZ Rebalance process should be suspended during instance replacement. By default, this value is set to true, meaning the ASG AZ Rebalance process is suspended during instance replacements. Setting this value to false will cause Instance Orchestrator to not suspend the AZ Rebalance process of the ASG during instance replacements.

Suspend ASG Termination

xosphere:instance-orchestrator:suspend-asg-termination

Flag indicating whether the Auto Scaling group Terminate process should be suspended during instance replacement. By default, this value is set to false, meaning the ASG Terminate process is not suspended during instance replacements. Setting this value to true will cause Instance Orchestrator to suspend the Terminate process of the ASG during instance replacements.

Terminate Instance SNS Topic

xosphere:instance-orchestrator:terminate-launch-topic-arn

The ARN of an SNS topic to publish a message whenever an instance is terminated. By default, this value is set to null indicating Instance Orchestrator will not publish any SNS messages when an instance is terminated.

Terminate Instance Error SNS Topic

xosphere:instance-orchestrator:terminate-error-topic-arn

The ARN of an SNS topic to publish a message whenever an instance termination fails. By default, this value is set to null indicating Instance Orchestrator will not publish any SNS messages when an instance termination fails.

Terminate On CodeDeploy Failure

xosphere:instance-orchestrator:terminate-on-codedeploy-failure

Flag indicating whether to terminate a replacement instance if the CodeDeploy deployment fails. By default, this value is set to true, meaning if a CodeDeploy deployment fails, the instance will be immediately terminated and a new instance will be launched to retry. Setting this value to false will ignore any CodeDeploy deployment failures.

Terraform AWS Provider Version

xosphere:instance-orchestrator:terraform-aws-provider-version

For Xogroups using Terraform integration, the version of the Terraform AWS Provider to use when updating the instance ID in the Terraform state file during an instance replacement.

Terraform Backend AWS Region

xosphere:instance-orchestrator:terraform-backend-aws-region

For Xogroups using Terraform integration, the AWS region that hosts the S3 bucket that stores the Terraform state file to use when updating the instance ID during an instance replacement.

Terraform Backend DynamoDB Table

xosphere:instance-orchestrator:terraform-backend-aws-region

For Xogroups using Terraform integration, the DynamoDB table to use for state file locking when updating the instance ID during an instance replacement.

Terraform Backend S3 Bucket

xosphere:instance-orchestrator:terraform-backend-s3-bucket-name

For Xogroups using Terraform integration, the S3 bucket that stores the Terraform state file to use when updating the instance ID during an instance replacement.

Terraform Backend S3 Key

xosphere:instance-orchestrator:terraform-backend-s3-key

For Xogroups using Terraform integration, the S3 key for the Terraform state file to use when updating the instance ID during an instance replacement.

Terraform Version

xosphere:instance-orchestrator:terraform-version

For Xogroups using Terraform integration, the version of the Terraform binary to use when updating the instance ID in the Terraform state file during an instance replacement.

Wait Period

xosphere:instance-orchestrator:wait-period-in-mins

Number of minutes to wait after an instance replacement is performed before replacing another instance. Accepts integer values. By default, this value is set to 0, meaning Instance Orchestrator will not wait any additional time before making a subsequent replacement. This setting can be useful for stateful application in Xogroups if a replacement would trigger an activity that may take some time such as a re-sync of nodes in a cluster (e.g. Hadoop, Cassandra, Elasticsearch, etc.).

Xogroup Enabled

xosphere:instance-orchestrator:xogroup-enabled

Flag indicating whether the Xogroup is enabled. By default, this value it set to true, indicating that the Xogroup is enabled, and Instance Orchestrator will perform instance replacements. If this value is set to false the Xogroup will be disabled and no instance replacements will occur.

Xogroup Spot Only

xosphere:instance-orchestrator:xogroup-spot-only

Flag indicating whether or not the Xogroup should use only Spot instances. By default, this value it set to false, indicating that if Spot instances are not available the Xogroup will replace them with On-Demand instances. If this value is set to true and Spot instances are not available, the Xogroup will not launch On-Demand instances and instead will wait for the Spot market to become available again.

Xogroup Thread Count

xosphere:instance-orchestrator:xogroup-thread-count

Number of threads to run in parallel when processing an Xogroup. Each thread will process one instance at a time. Accepts integer values. By default, this value is set to 1, meaning Instance Orchestrator will process one instance at a time. This setting is only applicable to Xogroups.

Upgrades

Frequency of Releases

Xosphere follows agile development methodologies and the frequency of changes will occur in congruence with agile methodologies. New releases will be immediate when patches are made, and continuous deployment will include the addition of minor new features added as frequently as daily although typically on a weekly cadence. Major releases are scheduled on a quarterly basis.

Upgrading with CloudFormation

To perform an update of the CloudFormation stack, navigate to the CloudFormation view in the AWS Console, locate and click on the xosphere-instance-orchestrator Stack.

Click the Update button, and choose the Replace current template radio button. In the Amazon S3 URL textbox enter the following URL:
https://xosphere-io-public-cfn-templates.s3-us-west-2.amazonaws.com/xosphere-instance-orchestrator/latest

Click the Next button, and complete the Update wizard.

Upgrading with Terraform

When using Terraform, to perform an upgrade of Xosphere Instance Orchestrator simply run:

```
terraform init
terraform apply
```

If using a pinned version of the Terraform module, be sure to update the version number in your Terraform script before running.

Container Orchestration Integration

Instance Orchestrator integrates natively with container orchestration platforms including: ECS, EKS, and Kubernetes (vanilla). Whenever Instance Orchestrator replaces an On-Demand instance with a Spot instance, or when it processes a Spot termination notification, all Pods or Tasks will be gracefully drained on the Node or Container Instance before it is shutdown.

In the case of ECS, no additional steps need to be taken in order to enable this integration. Instance Orchestrator will automatically detect if an instance is a Container Instance and drain its Tasks appropriately.

In the case of EKS or Kubernetes (vanilla), an additional installation step needs to take place and two Pods need to be deployed to your Kubernetes cluster.

The Xosphere Instance Orchestrator k8s Terminator is responsible for draining Kubernetes nodes when a Spot termination notification is received. The k8s Terminator runs a pod on each node in the cluster as part of a DaemonSet in the kube-system namespace. The k8s Terminator can safely be installed in any k8s cluster including managed services such as Amazon EKS.

The Xosphere Instance Orchestrator Instance Monitor is responsible for draining Kubernetes nodes when a replacement is taking place. The k8s Instance Monitor runs a pod on each node in the cluster as part of a DaemonSet in the kube-system namespace. The k8s Instance Monitor can safely be installed in any k8s cluster including managed services such as Amazon EKS.

Installation

Note:
In order for the Instance Orchestrator Lambda functions to be able to access your Kubernetes cluster, you must specify a subnet IDs and security group IDs in the CloudFormation parameters or Terraform configuration when installing Instance Orchestrator. If you did not select these options when installing, you can modify the stack and re-run it to ensure Instance Orchestrator is able to connect to your Kubernetes cluster.

To install the k8s Instance Monitor simply run the following kubectl command:

```
kubectl create -f https://xosphere-io-public-k8s-manifests.s3-us-west-
2.amazonaws.com/xosphere-k8s-instancemon.yaml
```

Expected Output:

```
clusterrole.rbac.authorization.k8s.io/xosphere-k8s-instancemon created
clusterrolebinding.rbac.authorization.k8s.io/xosphere-k8s-instancemon created
serviceaccount/xosphere-k8s-instancemon created
daemonset.extensions/xosphere-k8s-instancemon created
```

Validation

You can validate the k8s Instance Monitor is running on each node by getting the pods in the kube-system namespace:

```
kubectl get pods --namespace=kube-system
```

Expected Output:

```
NAME                                        READY   STATUS    RESTARTS   AGE
xosphere-k8s-instancemon-ey6ws              1/1     Running   0          1m
xosphere-k8s-instancemon-9etnb              1/1     Running   0          1m
xosphere-k8s-instancemon-a2sdf              1/1     Running   0          1m
```

Note:
You will have other pods running in the kube-system namespace in addition to the xosphere-k8s-instance-monitor-xxxxx Pods. You should see three xosphere-k8s-instance-monitor-xxxxx Pods in the cluster.

Logs

To view the logs for a given k8s Instance Monitor pod, simply run the following command (be sure to replace the xxxxx with the correct randomized extension listed in the get pods output):

```
kubectl logs xosphere-k8s-instancemon-xxxxx --namespace=kube-system
```

Uninstall

In the event that you wish to uninstall the k8s Instance Monitor from your cluster, simply run the following commands:

```
kubectl delete -f https://xosphere-io-public-k8s-manifests.s3-us-west-
2.amazonaws.com/xosphere-k8s-instancemon.yaml
```

Expected Output:

```
clusterrole.rbac.authorization.k8s.io "xosphere-k8s-instancemon" deleted
clusterrolebinding.rbac.authorization.k8s.io "xosphere-k8s-instancemon" deleted
serviceaccount "xosphere-k8s-instancemon" deleted
daemonset.extensions "xosphere-k8s-instancemon" deleted
```

Disaster Recovery

Xosphere Instance Orchestrator does not store any state with the AWS account. This means that disaster recovery is as easy as re-installing the CloudFormation stack or Terraform module.

Support

Xosphere provides Standard Support during business hours for all customers at no additional cost. Premier Support is also available for an additional fee. To increase your support level please contact your Xosphere representative or sales@xosphere.io.

To open a support ticket, you can contact the Xosphere Support team by email at support@xosphere.io.

Telephone support is available to Premier Support customers and can be reached by calling 800-905-4125. Chat support is also available to Premier Support customers. For more details about accessing the chat support channel please contact your Xosphere representative.