

Reservation Capacity Slack Alerts

1. Versions

Version	Date	User	Comment
0.0.1	Sep 27, 2018	administra tor	
0.0.2	Sep 27, 2018	administra tor	

Current version:0.0.2

2. Outputs

Name	Type	Description
statusCodeToken	number	Status code of the HTTP request
contentLengthToken	number	Length of the request response
headersToken	Properties	Response headers

3. Attributes

Name	Type	Description
restOperationToken	REST:RESTOperation	REST operation to be used for the call
defaultContentType	string	Input content type
contentToken	string	Input content
contentAsStringToken	string	Response content as string
tokenId	string	
restOperationReservatio n	REST:RESTOperation	REST operation to be used for the call
thresholdMemAlloc	number	Threshold for memory allocation as a percentage
thresholdStorAlloc	number	Threshold for storage allocation as a percentage

Name	Type	Description
thresholdStorUsed	number	Threshold for storage used as a percentage
slackWorkflowId	string	Workflow ID of the sendSlackMessage workflow
vcacHostId	string	ID of the vCAC Host object to check reservations

4. Parameter presentation

Parameter	Description	Parameter Info
-----------	-------------	----------------

5. Workflow schema



6. Workflow items

Name	Type	Next item	Description
	WorkflowItemEnd		
Acquire Token	WorkflowItemLink	Process data and send message	Invokes a REST operation.
Process data and send message	WorkflowItemTask		

7. Source code for the used actions

7.1. Process data and send message

```
// Setup Slack items
var slackwf = Server.getWorkflowWithId(slackWorkflowId);
var slackwfProperties = new Properties();

// Extract Token's id value
var token = JSON.parse(contentAsString)
tokenId = token.id;

headerParam_0 = "Bearer " + tokenId;

// Testing
System.log("Starting testing output...");
System.log("\tThreshold for Memory Allocation is: " + thresholdMemAlloc + "%");
System.log("\tThreshold for Storage Allocation is: " + thresholdStorAlloc + "%");
System.log("\tThreshold for Storage Used is: " + thresholdStorUsed + "%");

// Connect to vRA host and get reservation info
var vcacCafeHost = Server.findForType("vCACCAFE:VCACHost",vcacHostId);
var endpoint = 'com.vmware.vcac.core.cafe.reservation.api';
var restClient = vcacCafeHost.createRestClient(endpoint);
var resItemsUrl = "reservations/info";
var resItems = restClient.get(resItemsUrl);
var response = resItems.getBodyAsString();
var json = JSON.parse(response);
var content = json.content;

// iterate through contents
for each(var x in content) {
    // Set flag to false at the start of each reservation
    var thresholdFlag = false;
    var tmpAlertMessage = "";
    // Keys:
    @type,id,name,reservationTypeRef,tenantId,subTenantRef,enabled,priority,reservationPolicyRef,extensionData
    var reservationEntries = x.extensionData.entries;
    for each(var y in reservationEntries) {
        // Init the variables for temp storage of items
        var tmpResEndpoint = "";
        var tmpResComputeResource = "";
        var tmpResMemAlloc = "";
        var tmpResStorAlloc = "";
        var tmpResStorUsed = "";
        var reservationKey = y.key;
        var reservationValue = y.value.value; // The actual value is nested, hence the double
        value
    }
}
```

```
// Start switch logic
switch(reservationKey) {
    case "endpoint":
        // The endpoint name
        System.log("\tEndpoint: " + reservationValue);
        tmpResEndpoint = reservationValue;
        break;
    case "computeResource":
        // name of the host or cluster?
        System.log("\tCluster or host: " + reservationValue);
        tmpResComputeResource = reservationValue;
        break;
    case "memoryAllocatedPercentage":
        // Memory consumption in format of: xx% (y GB of z GB)
        tempMemAlloc = reservationValue.split("%");
        tmpResMemAlloc = parseInt(tempMemAlloc[0]);
        if (tmpResMemAlloc >= thresholdMemAlloc) {
            // High memory condition
            System.log("Too much memory allocated, sending slack message!");
            // Setup Slack items
            var slackwfMemAlloc = Server.getWorkflowWithId(slackWorkflowId);
            var slackwfMemAllocProperties = new Properties();
            var slackMessageMemAlloc = ":rotating_light: WARNING: *" + tmpResMemAlloc +
"%* memory allocation on reservation `" + x.name + "`! :rotating_light:";
            slackwfMemAllocProperties.put("slackMessage",slackMessageMemAlloc);
            var slackwfMemAllocToken =
slackwfMemAlloc.execute(slackwfMemAllocProperties);
            thresholdFlag = true;
        } else {
            System.log("Still under threshold");
        }
        System.log("\tCurrent memory allocation consumption is: " + tmpResMemAlloc + "
(vs threshold of " + thresholdMemAlloc + ")");
        break;
    case "storageAllocatedPercentage":
        // Storage allocation consumption in format of: xx% (y GB of z GB)
        tempStorAlloc = reservationValue.split("%");
        tmpResStorAlloc = tempStorAlloc[0];
        if (tmpResStorAlloc >= thresholdStorAlloc) {
            // High Storage Allocation condition
            System.log("Too much storage allocated, sending slack message!");
            // Setup Slack items
            var slackwfStorAlloc = Server.getWorkflowWithId(slackWorkflowId);
            var slackwfStorAllocProperties = new Properties();
            var slackMessageStorAlloc = ":rotating_light: WARNING: *" + tmpResStorAlloc
```

```
+ "%* storage allocation on reservation `" + x.name + "`! :rotating_light:";
    slackwfStorAllocProperties.put("slackMessage",slackMessageStorAlloc);
    var slackwfStorAllocToken =
slackwfStorAlloc.execute(slackwfStorAllocProperties);
    thresholdFlag = true;
}
    System.log("\tCurrent storage allocation consumption is: " + tmpResStorAlloc);
    break;
case "storageUsedPercentage":
    // Storage usage consumption in format of: xx% (y GB of z GB)
    tempStorUsed = reservationValue.split("%");
    tmpResStorUsed = tempStorUsed[0];
    if (tmpResStorUsed >= thresholdStorUsed) {
        // High Storage Used condition
        System.log("Too much storage used, sending slack message!");
        // Setup Slack items
        var slackwfStorUsed = Server.getWorkflowWithId(slackWorkflowId);
        var slackwfStorUsedProperties = new Properties();
        var slackMessageStorUsed = ":rotating_light: WARNING: *" + tmpResStorUsed +
"%* storage used on reservation `" + x.name + "`! :rotating_light:";
        slackwfStorUsedProperties.put("slackMessage",slackMessageStorUsed);
        var slackwfStorUsedToken =
slackwfMemAlloc.execute(slackwfStorUsedProperties);
        thresholdFlag = true;
    }
    System.log("\tCurrent storage used consumption is: " + tmpResStorUsed);
    break;
default:
    // Other items we don't care about
}
}
System.log("Threshold flag is: " + thresholdFlag);
if (thresholdFlag == true) {
    System.log("A threshld was breached for entry blah, send an alert to slack");
}
System.log("\n");
}
System.log("Finished testing output");
```