OmniDJ Cloud - Dynamic Cloud Allocator overall algorithm

```
def start_gpu_annotation_process(self, file_path, file_type, computed_eta, rotate=False, gdpr=True):
    """ Run VAPOR engine on a Azure VM machine """
    start = time()
    print('Start GPU task')
    print('File {}'.format(file_path))

    task_id = self.request.id

    file_name = path.basename(file_path)

    if file_type == 'image':
        self.start_cpu_annotation_process(file_path, file_type, computed_eta, rotate=rotate)
    elif file_type != 'video':
        # Raise for any file other than image and video
        raise ValueError('File should be video or image, not {}'.format(file_type))

    ws_send_next_stage(task_id, 'loading', 12.1)

    r = redis.Redis(host=settings.IDLE_HOST, port=settings.IDLE_PORT, db=settings.IDLE_DB_NUMBER)
    idle_expire_time = settings.IDLE_EXPIRE_TIME

    if not path.exists(media_res_path):
        makedirs(media_res_path)

    # Get available Azure VMs
    azure_vm = AzureVM(media_src_path, media_res_path, file_type, rotate, task_id=task_id)
    chosen_vm = azure_vm._get_available_vm()

    retries = 0
    while not chosen_vm and retries < 30:
        sleep(10)
        chosen_vm = azure_vm._get_available_vm()

    ws_send_next_stage(task_id, 'loading', 41.62)

    redis_key = '{account}__{group}__{vm}'.format(account=chosen_vm['AZURE_ACCOUNT_NAME'],
                                                  group=chosen_vm['GROUP_NAME'],
                                                  vm=chosen_vm['VM_NAME'])

    vm_info_string = '{user}__{ip}__{passwd}__{redis_key}'.format(user=chosen_vm['VM_USERNAME'],
                                                                  ip=chosen_vm['VM_IP'],
                                                                  passwd=chosen_vm['VM_PASSWORD'],
                                                                  redis_key=redis_key)
    UploadModel.objects.filter(original_file=file_path).update(vm_info_string=vm_info_string)

    # Start Azure VM
    chosen_vm, vm_idle = azure_vm.start_vm(chosen_vm)

    ws_send_next_stage(task_id, 'loading', 98.37)
```