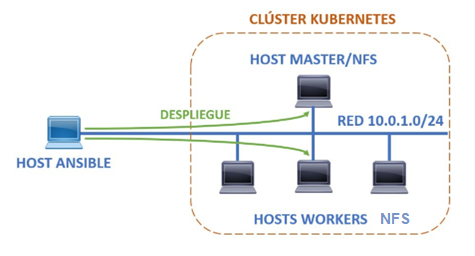
Automatización de despliegue

El repositorio donde está ubicado esta práctica es el siguiente: ***https://github.com/Markitoxrr/CP2/tree/master***

Para la realización de esta práctica he elegido desplegar la aplicación Jenkins.

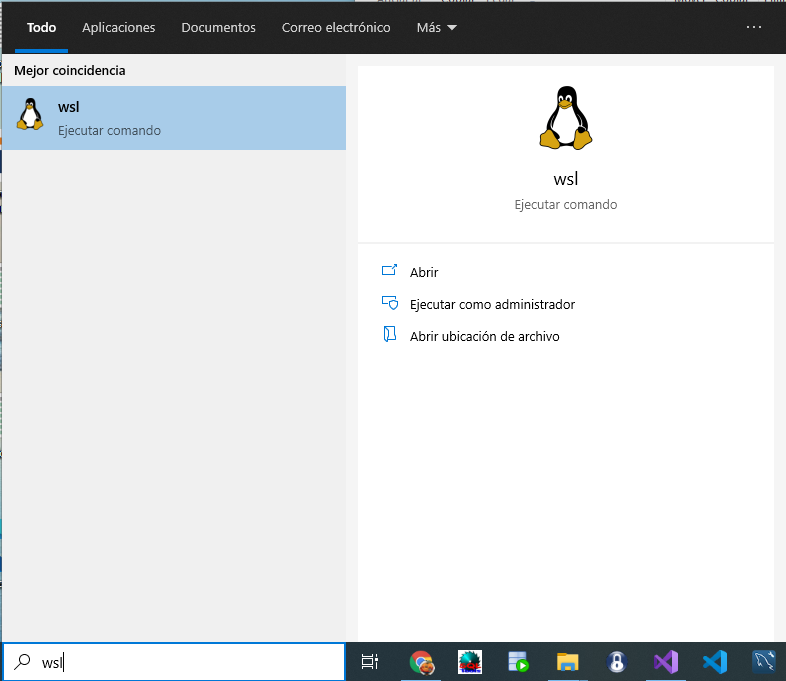
Para el despliegue se utilizarán 4 equipos en clúster con un equipo master, 2 workers y un nfs. Todos los equipos tendrán CentOS 8 y una configuración con 2 CPU y 4 GB de RAM.



Para la instalación de la aplicación vamos a utilizar Terraform para desplegar la infraestructura necesaria en Azure y Ansible para la instalación de los paquetes necesarios en los servidores, así como, la configuración del clúster e instalación de la propia aplicación.

1. **Despliegue de la infraestructura con Terraform**

Todo lo necesario para desplegar la infraestructura con Terraform se encuentra dentro de la carpeta ***terraform*** dentro del directorio CP2.

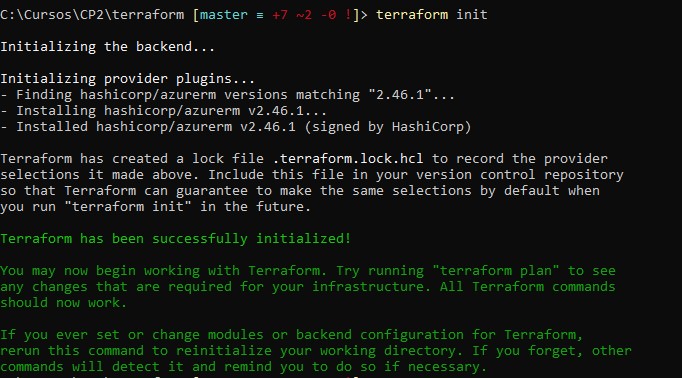


* 1. Lo primero que tenemos que hacer es abrir un terminal de Linux, en mi caso, como dispongo de Windows 10, he utilizado wsl e ir a la ruta donde se encuentra el código de Terraform, en mi caso ***/mnt/c/Cursos/CP2/terraform.***

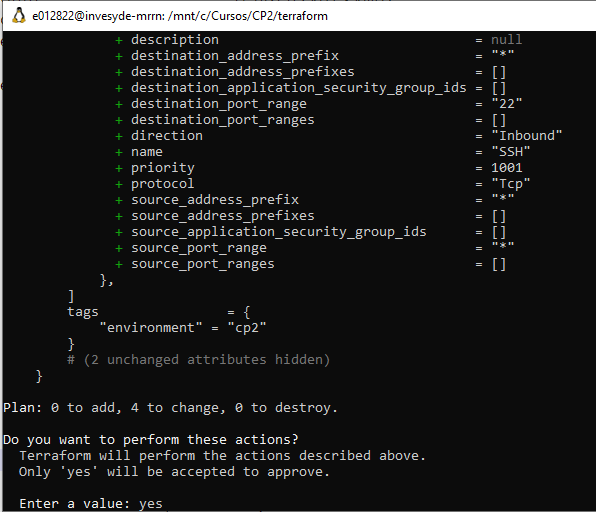
Esto abrirá una consola de comandos donde podremos ir escribiendo todos los comandos necesarios.

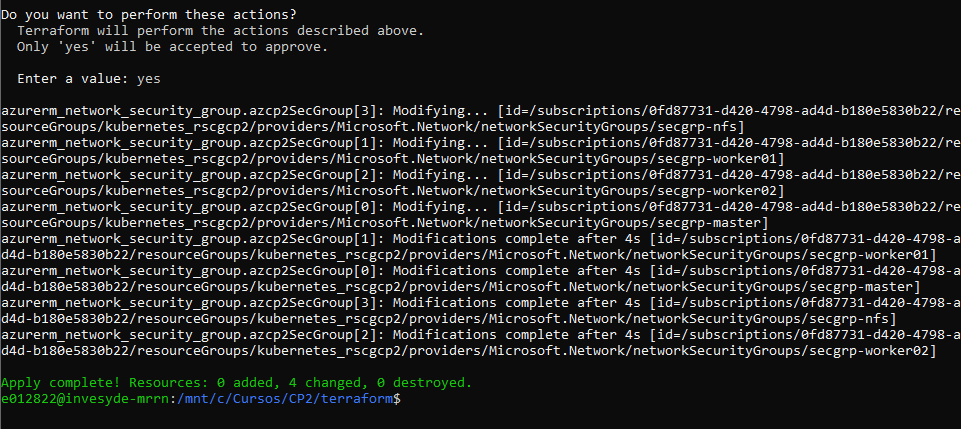


* 1. Lo primero que deberos hacer es iniciar Terraform con el comando ***“terraform init”*.** Esto creará los ficheros terraform.tfstate, terraform.tfstate.backup , .terraform.lock.hcl y la carpeta .terraform dentro de la carpeta Terraform.

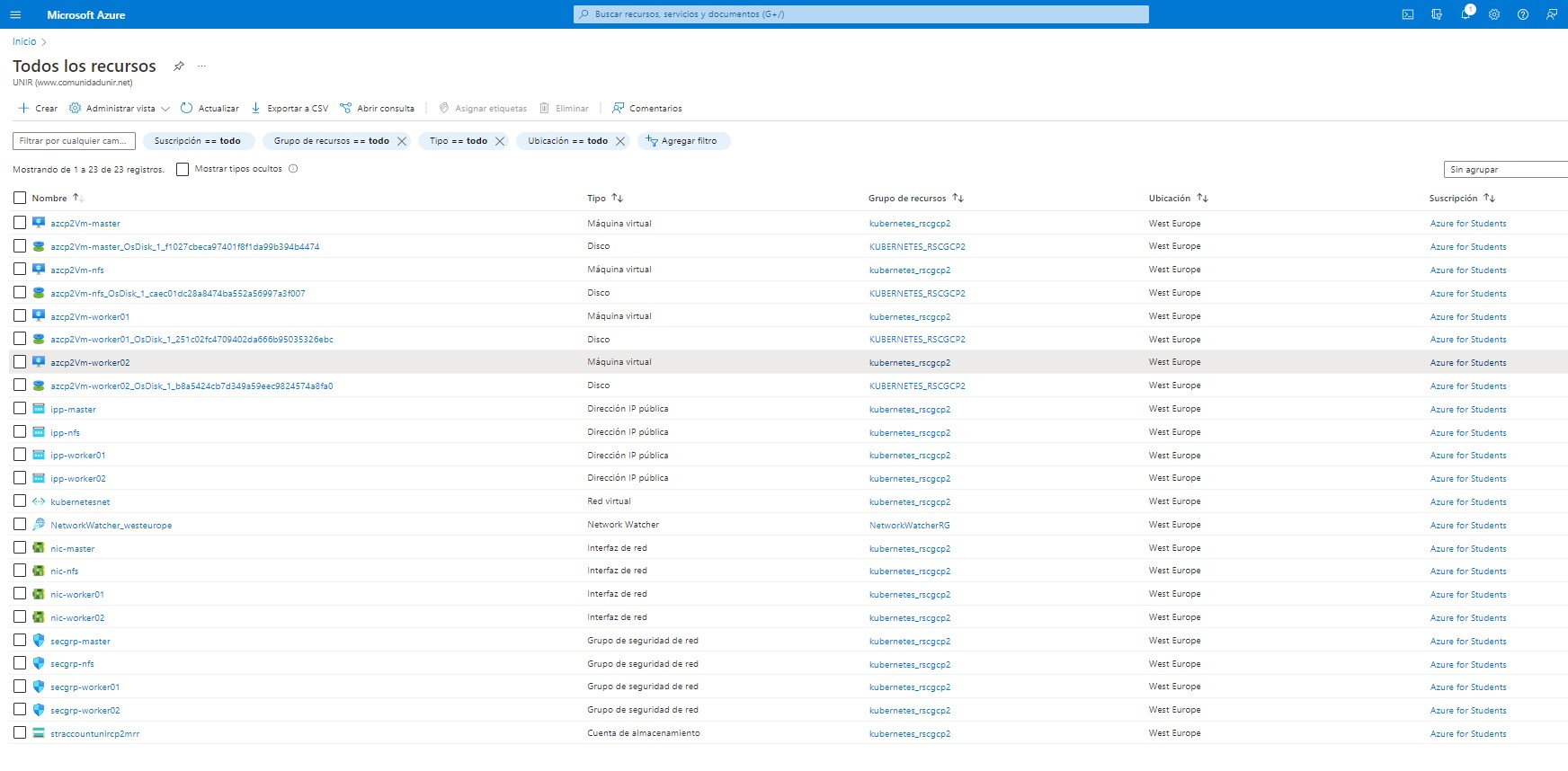


* 1. Ahora debemos escribir el comando “terraform apply” para que terraform despliegue toda la infraestructura en Azure. Lo primero que hace Terraform es analizar si todo el contenido de los ficheros de despliegue es correcto, si es así, nos preguntará si queremos realizar en despliegue. Escribimos “yes”.





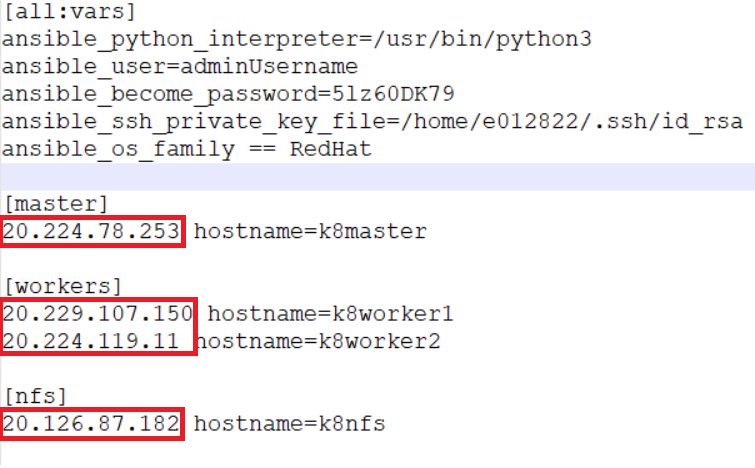
* 1. Una vez finalizado el despliegue, podemos ir a nuestra cuenta de Azure a la zona de recursos y comprobar que se han desplegado todos los recursos **que definimos.**



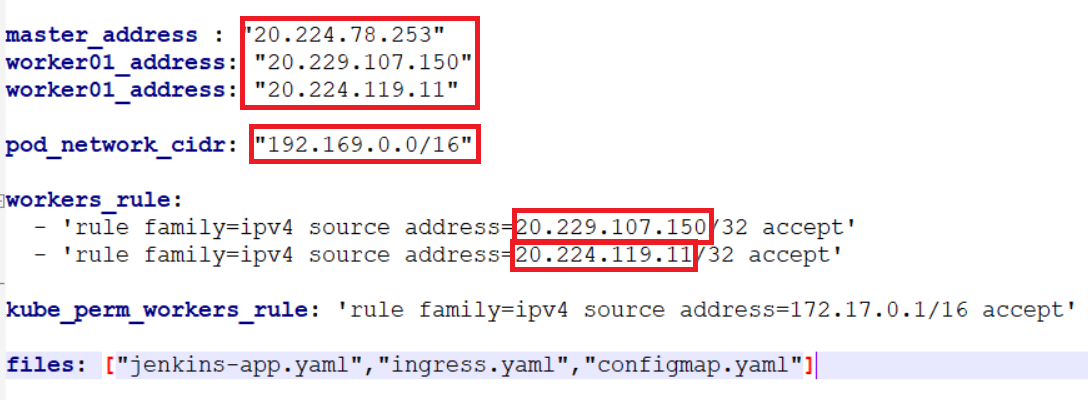
1. **Despliegue de aplicación con Ansible**

Todo lo necesario para desplegar la aplicación con Ansible se encuentra dentro de la carpeta ***ansible*** dentro del directorio CP2

* 1. Lo primero que debemos hacer es modificar los ficheros ***/mnt/c/Cursos/CP2/ansible/hosts*** y ***/mnt/c/Cursos/CP2/ansible/groups\_vars/all.yaml*** para añadir las IP públicas de los servidores desplegados obtenidas desde Azure.

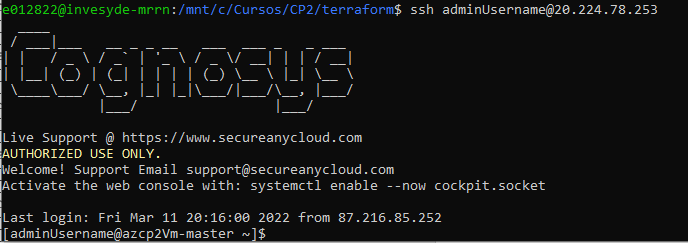


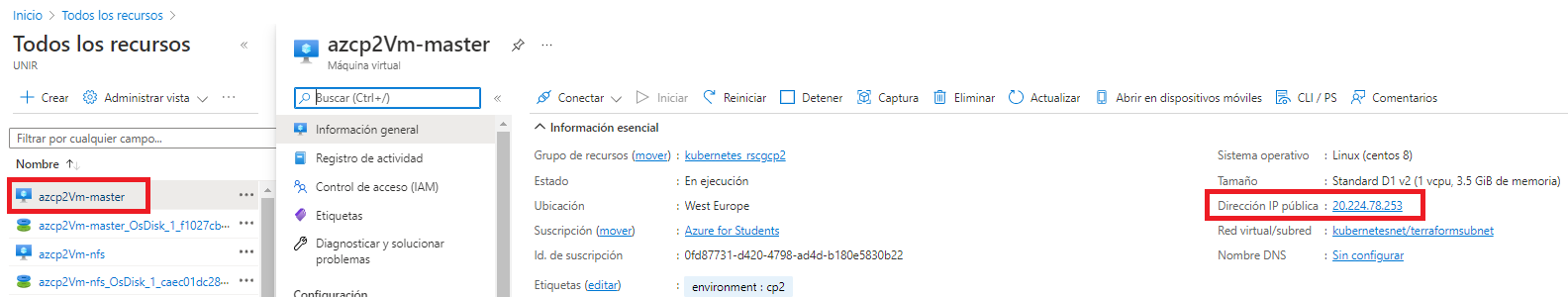
**Fichero C:\Cursos\CP2\ansible\Hosts**



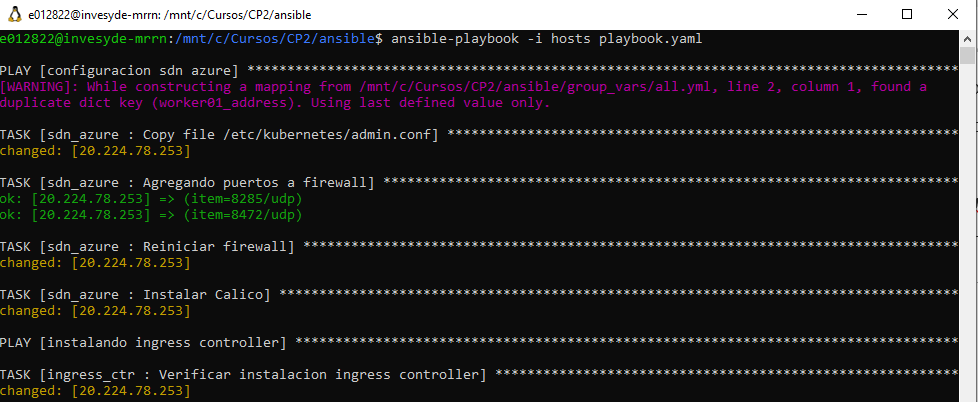
**Fichero C:\Cursos\CP2\ansible\group\_vars\all.yaml**

* 1. Lo primero que debemos hacer es acceder a todas las máquinas (Master, Worker1, Worker2 y Nfs) una por una mediante el comando ***ssh adminUsername@[Ip pública de la máquina].*** La Ip pública de cada máquina se puede encontrar en Azure, dentro de Todos los Recursos y accediendo a cada máquina para que no nos de error a la hora de desplegar

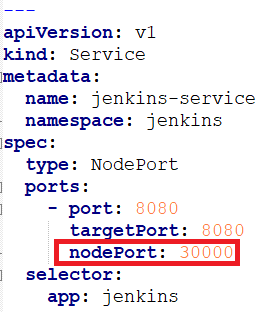


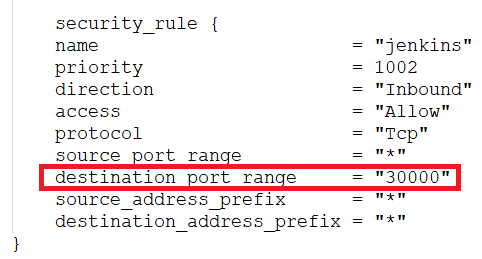


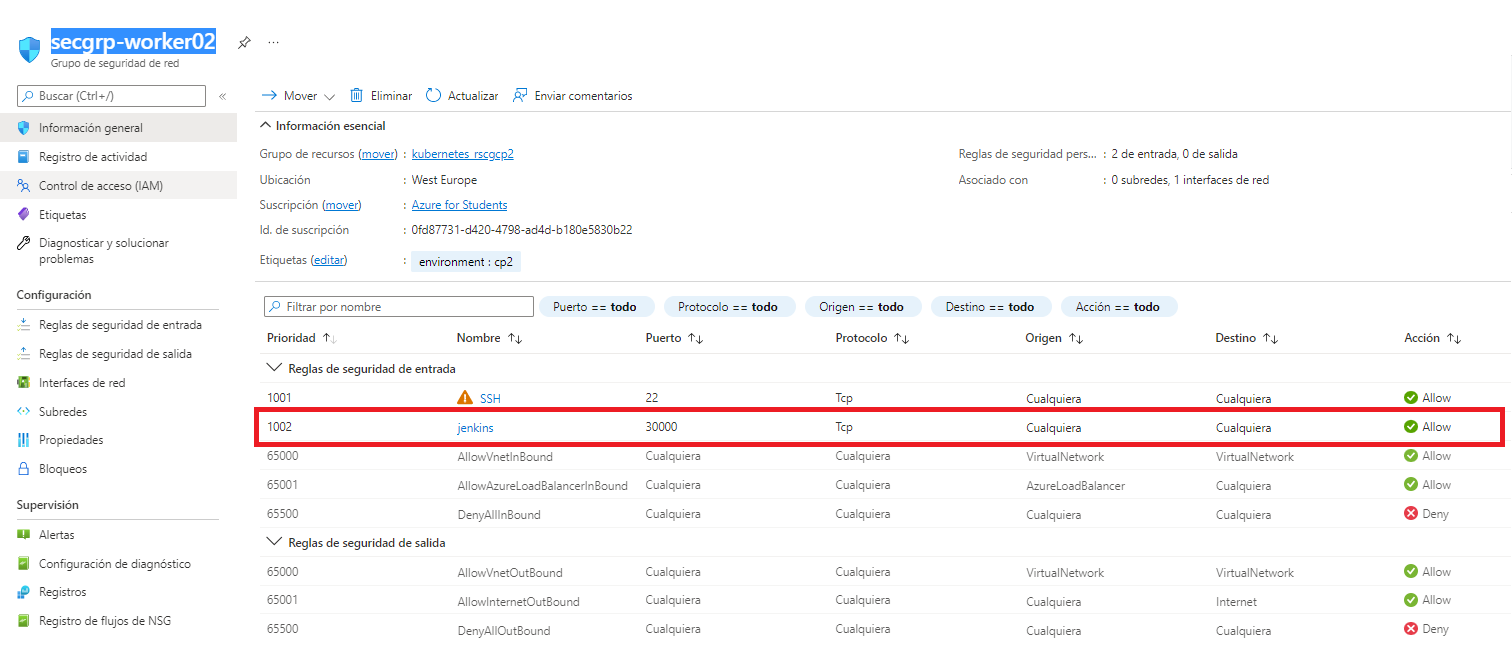
* 1. Nos situamos, con wsl, dentro del directorio **/mnt/c/Cursos/CP2/ansible.**
  2. Ahora hay que ejecutar el ***comando ansible-playbook -i hosts playbook\_cp2.yaml*** y automáticamente empezará a desplegarse todo lo necesario para preparar los servidores, crear el clúster e instalar la aplicación.



* 1. Si todo ha ido bien, una vez desplegada la aplicación, tenemos que abrir el puerto por el que vamos a acceder a la misma desde fuera. Para ello debemos añadir una security\_rule en el fichero ***CP2/terraform/security.tf*** abriendo el puerto 30000 que es donde hemos desplegado Jenkins.







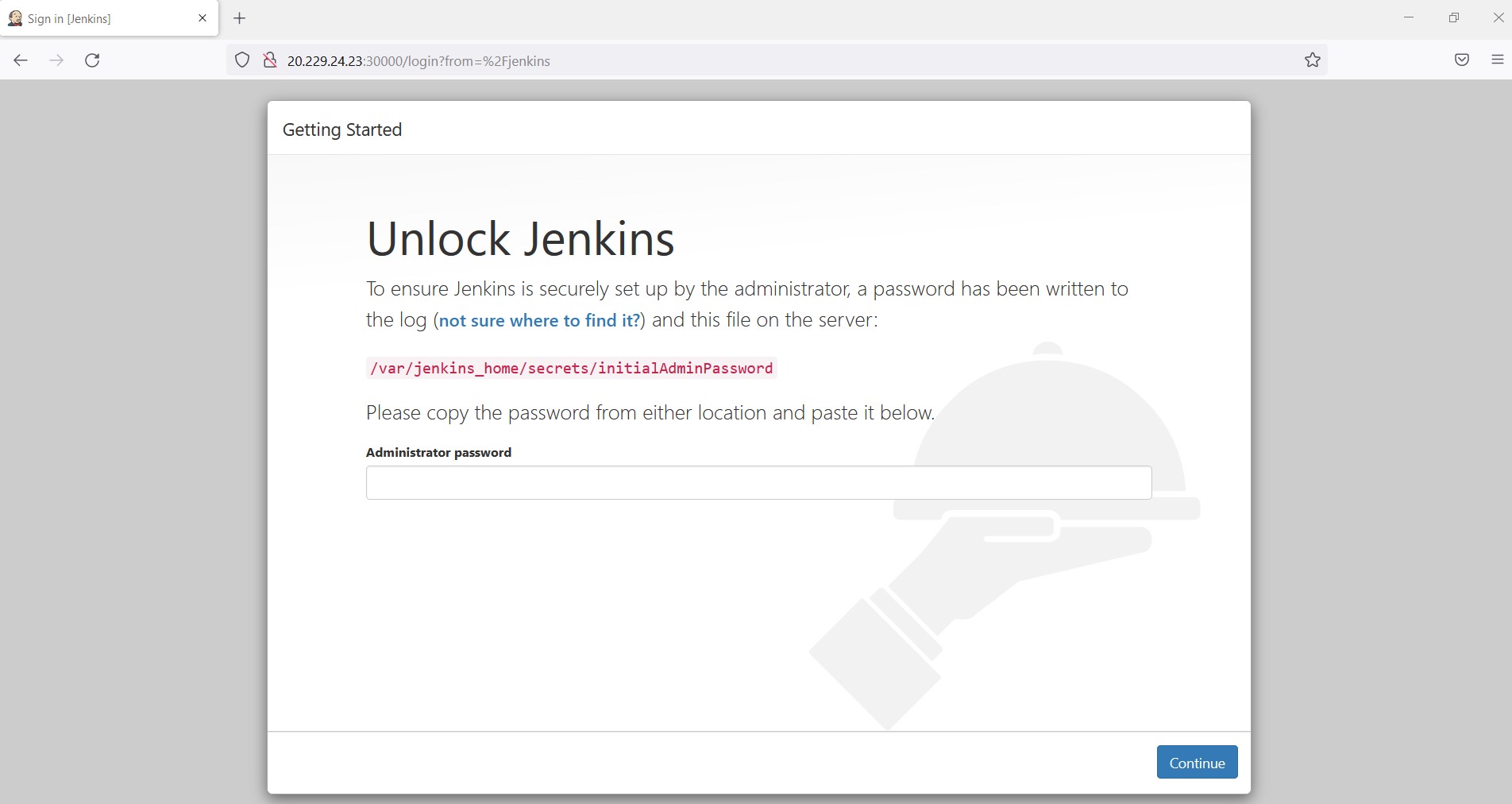
**Security\_rule para abrir el puerto de acceso a la aplicación**

Podemos comprobar que el puerto se ha abierto correctamente yendo a nuestra cuenta de Azure, Todos los recursos y ahí buscando el Security Group para uno de los servidores, por ejemplo el Worker 2 y comprobando que aparece el puerto 30000 como permitido.

**Servicio donde se despliega la aplicación**

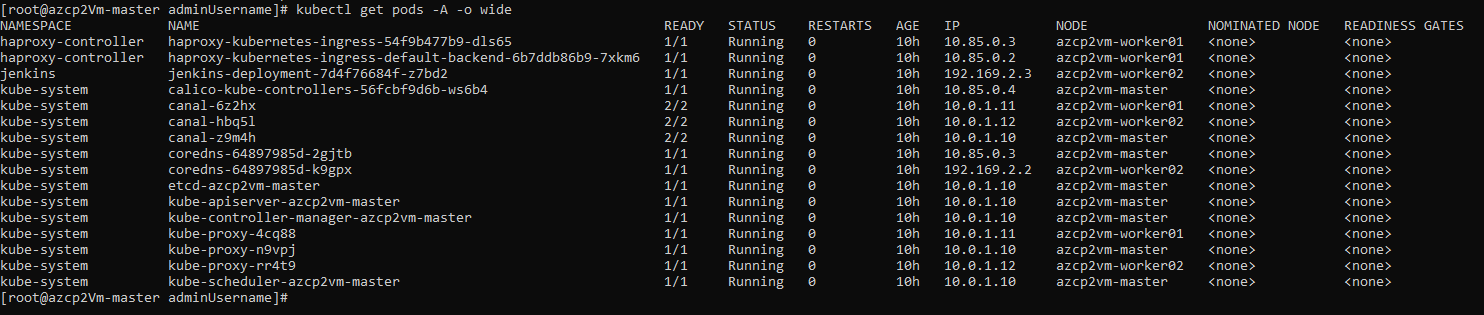
* 1. Para comprobar que el despliegue ha sido correcto, podemos escribir en un

navegador la dirección pública de algunos de los Workers seguido del puerto donde corre la aplicación. Para el caso de nuestro ejemplo, sería en el Worker 2: <http://20.229.24.23:30000/>.

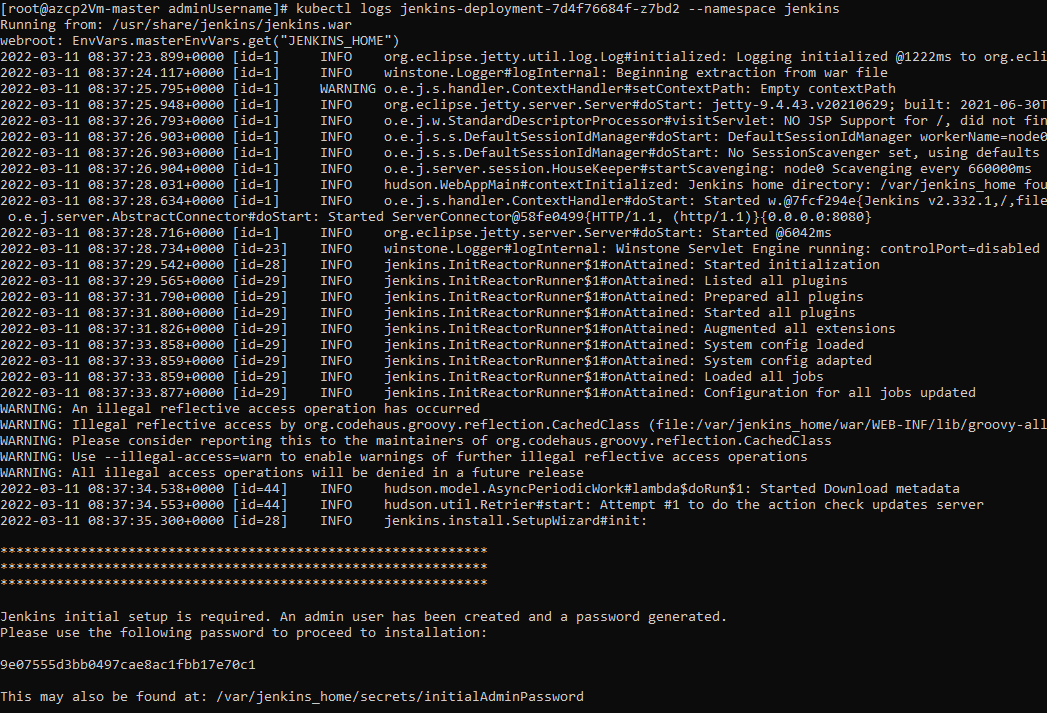


* 1. Para obtener la clave de desbloqueo de Jenkins tenemos que acceder a la máquina master, cambiar a ***root*** con el comando ***sudo su*** y mostrar todos los pods con el comando kubectl get pods –A –o wide.

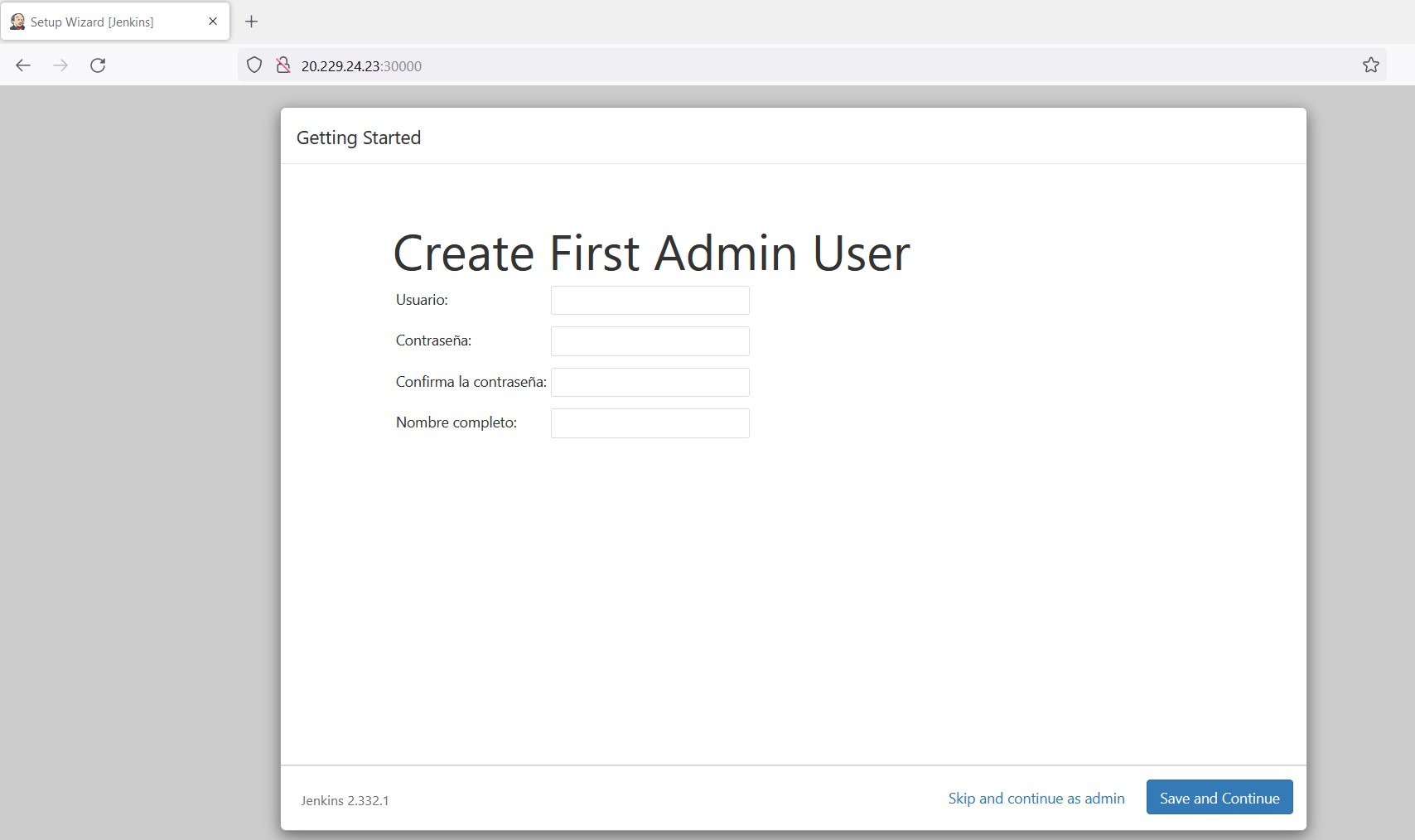
Con este comando podemos obtener el nombre del pod donde está corriendo Jenkins: ***jenkins-deployment-7d4f76684f-z7bd2***

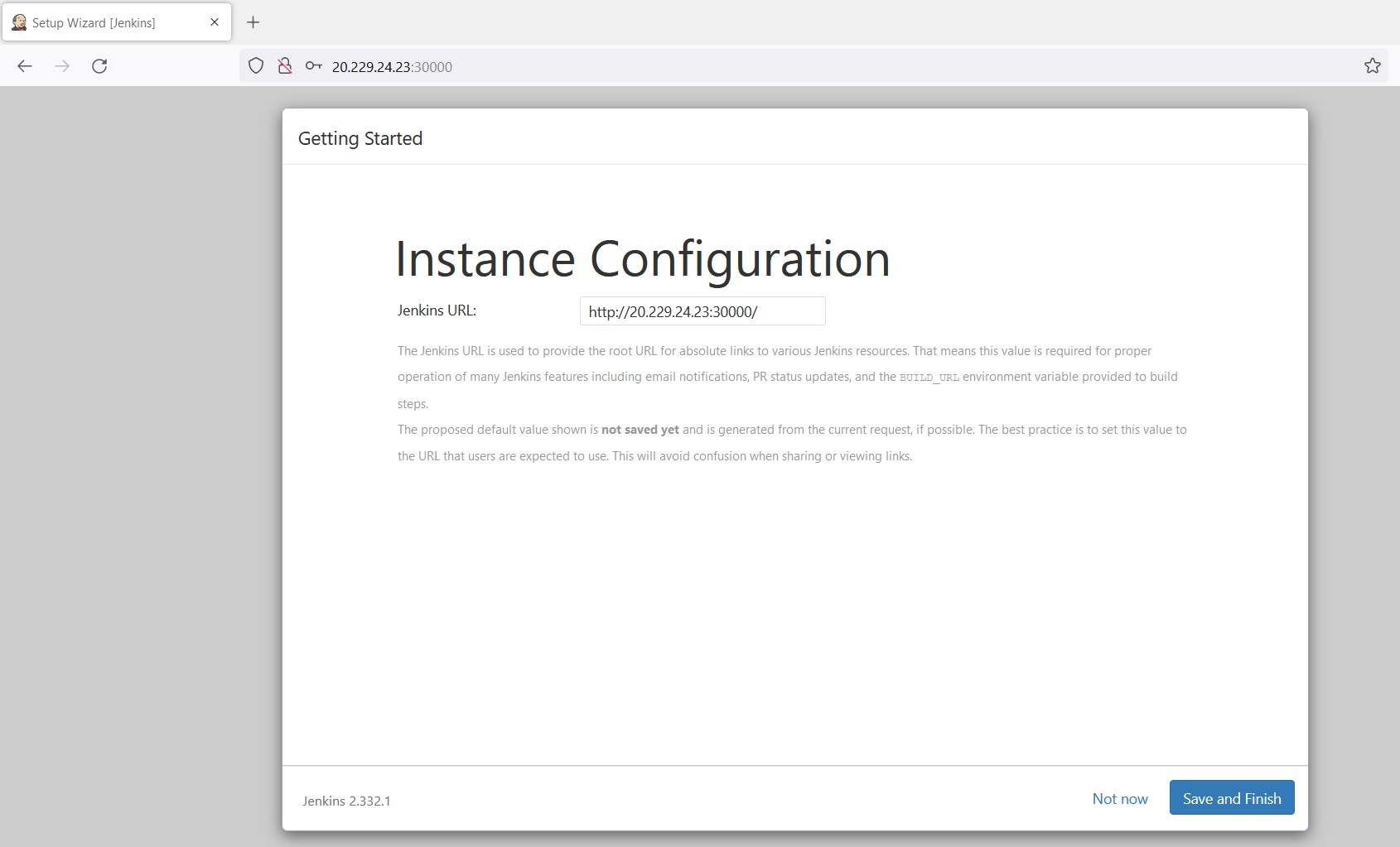


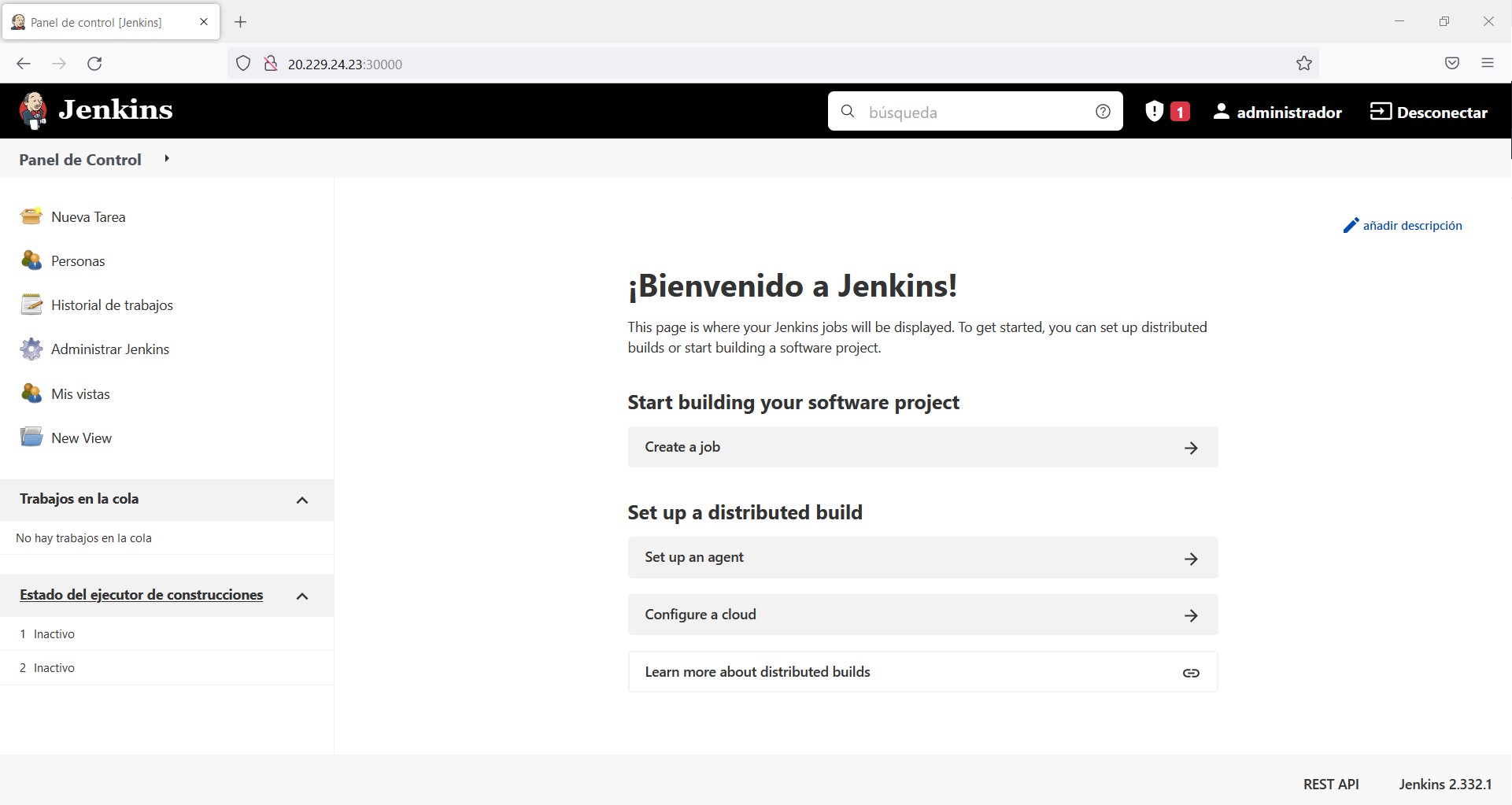
* 1. Una vez obtenido el nombre del pod, ejecutamos el siguiente comando para obtener la clave de administrador que nos permitirá desbloquear Jenkins: ***kubectl logs jenkins-deployment-7d4f76684f-z7bd2 --namespace Jenkins***



* 1. Una vez obtenida la calve podemos introducirla en la web de Jenkins y continuar configurándolo.







1. **Problemas encontrados durante el desarrollo de la práctica**

Para ser sinceros el principal problema que me he encontrado durante el desarrollo de la práctica he sido yo mismo y mi desconocimiento de Linux.

He de agradecer sinceramente la gran paciencia y labor desarrollada por mi profesor José Ángel de Bustos a la hora de ayudarme y responder a todas mis preguntas en los diferentes foros.

Durante el desarrollo de la práctica me he encontrado gran diversidad de errores que pasaré a enumerar:

* Primer problema, mi equipo es Windows 10 con 8GB no tenía recursos para poder desplegar el laboratorio como se aconsejaba en la práctica, por lo que tuve que utilizar ***wsl*** y acostumbrarme a ello.
* En la parte de Terraform no he encontrado muchos problemas, de echo he podido desplegar todas las máquinas sin problema. También me ha dejado desplegar la aplicación sin ningún mensaje de error o capado por parte de Azure. Lo único que vi es que en el Worker 1 no pude acceder a Jenkins mientras que en el Worker 2 sí. No sé si esto será problema de Azure por limitaciones de la cuenta.
* Al desplegar por primera vez no me desplegaba la aplicación porque salía continuamente el mensaje de: ***Are yo usure you want to continue connecting (yes/no/[fingerprint])? Yes.*** Lo solucioné accediendo primeramente a las máquinas antes de desplegar.
* He tenido problemas con el fichero ***$HOME/.kube/config*** porque me duplicaba su contenido y lo ponía mal de tal forma que el fichero daba error de formato al leerlo.

TASK [sdn\_azure : Instalar Calico] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
fatal: [20.126.144.60]: FAILED! => {"changed": true, "cmd": ["kubectl", "apply", "-f", "https://docs.projectcalico.org/manifests/canal.yaml"], "delta": "0:00:00.053219", "end": "2022-03-05 17:23:26.642763", "msg": "non-zero return code", "rc": 1, "start": "2022-03-05 17:23:26.589544", "stderr": "error: error loading config file \"/root/.kube/config\": yaml: line 19: mapping values are not allowed in this context", "stderr\_lines": ["error: error loading config file \"/root/.kube/config\": yaml: line 19: mapping values are not allowed in this context"], "stdout": "", "stdout\_lines": []}

* El peor problema que me he encontrado, y con el que estuve lidiando varios días, fue que una vez desplegada la aplicación webapp-routed que puso el profesor de ejemplo en las clases, no me dejaba acceder.
  + Primero fue, porque se me había olvidado exponer el puerto en el que corre el pod mediante una nueva security\_rule.
  + Después pensábamos que era porque no había definido en el fichero ***/etc/hosts*** de mi equipo, una entrada para foo.bar con la ip del master.
  + Por último, después de la inestimable ayuda de José Ángel, se dio cuenta que no había desplegado correctamente el ingress\_controller porque estaba utilizando la versión v1.5 que daba problemas, así que, modifique el despliegue del ingress\_controller para que utilizase la versión v1.7 y ya se desplegó todo correctamente.