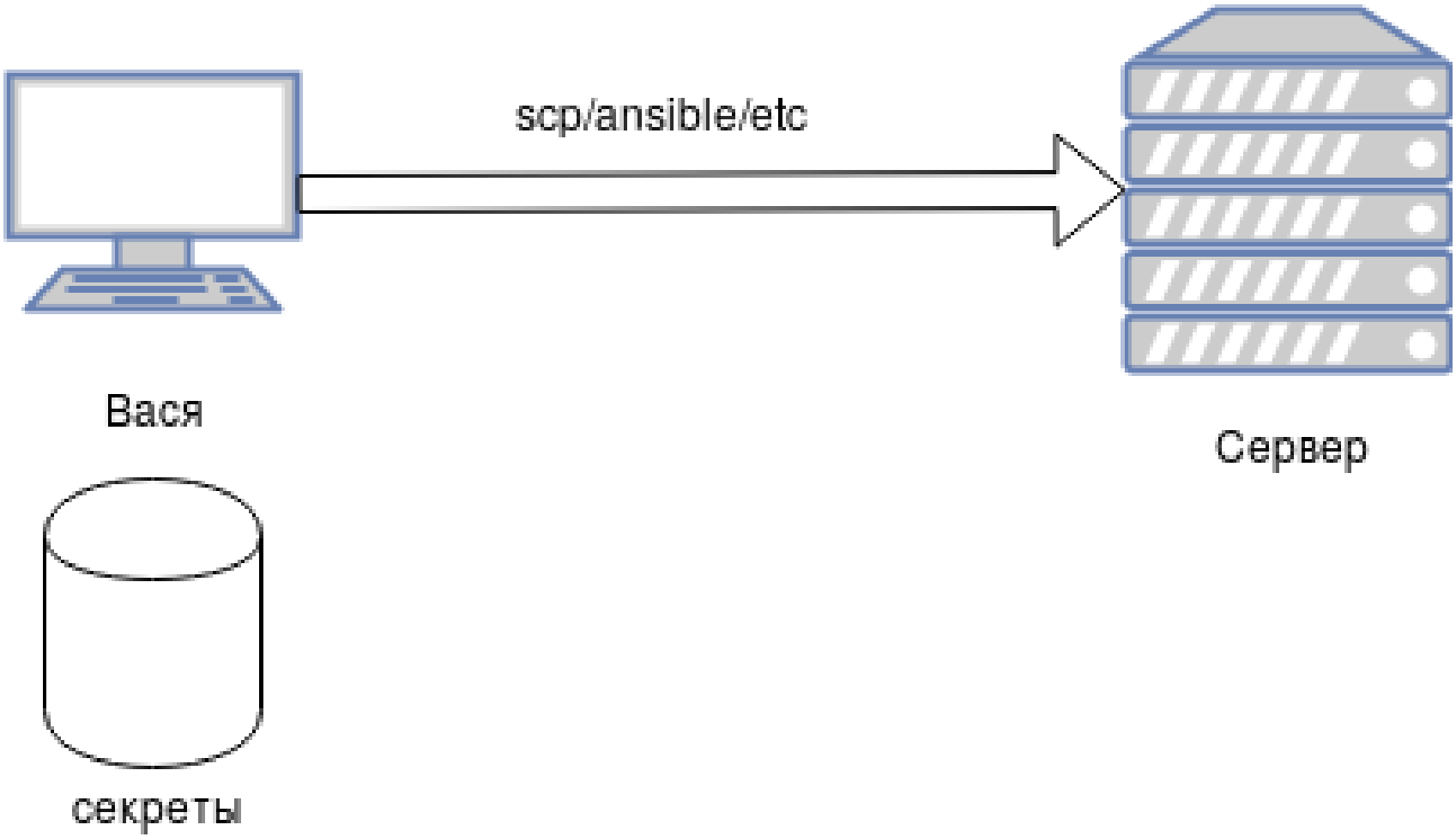
Безопасное хранение секретов

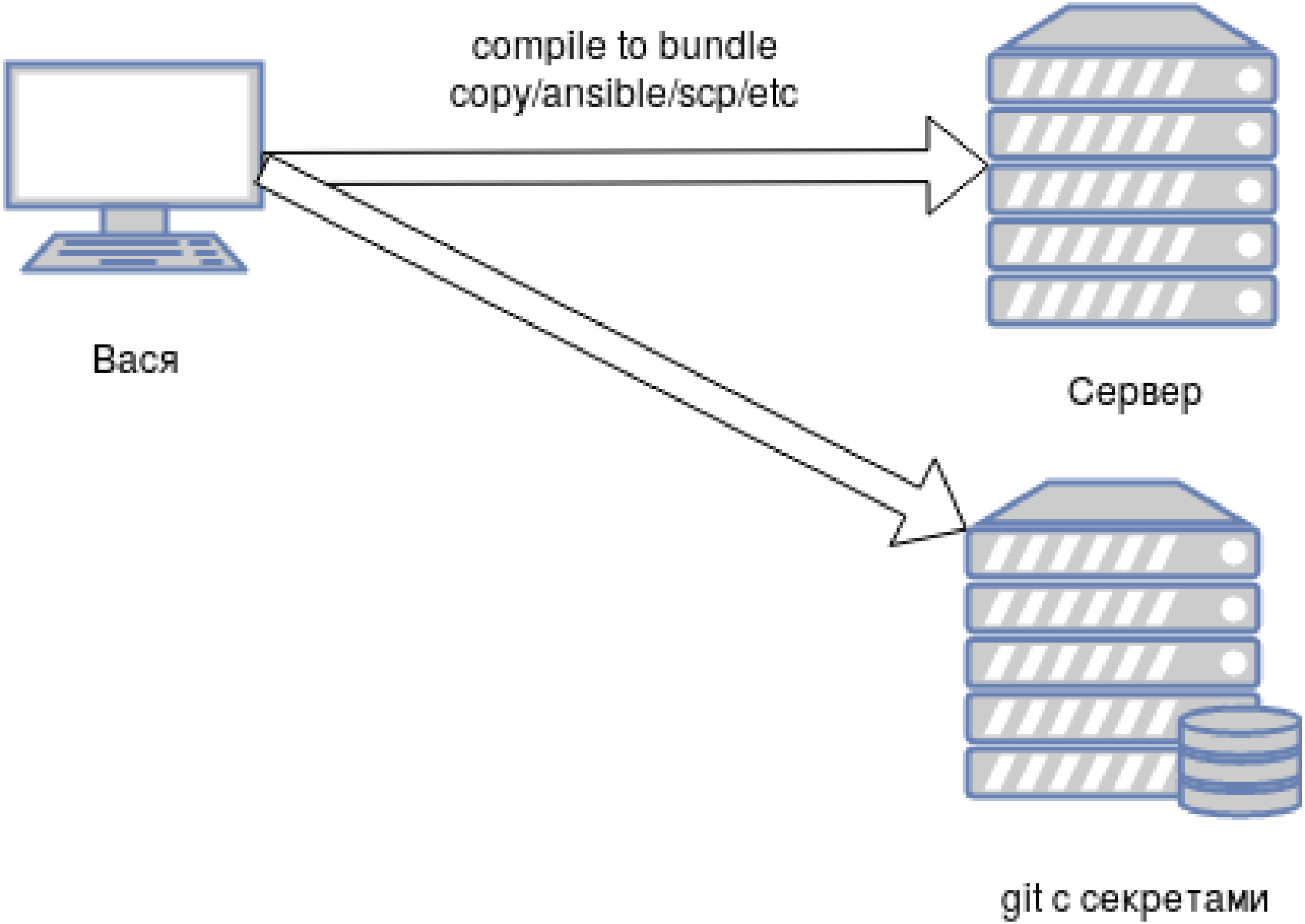
Опыт использования Hashicorp Vault

Безопасность - всему голова

Ключи непосредственно на машинах



Ключи в репозитории



И что с того?

- Человеческий фактор
- Неудобство использования
- Потенциальная уязвимость секретов

Давным-давно, при динозаврах

Hashicorp Vault

- Безопасное и надежное хранение
- Обновление секретов
- Аудит использования
- Генерация используя бэкенды
- Система плагинов

Хранилища

- Azure
- Consul
- Etcd
- In-Memory
- MySQL
- Zookeeper
- другие

Consul storage

- Регистрация в консуле из коробки
- Высокая доступность
- Данные в консуле в зашифрованном виде
- Официальная поддержка
- У нас уже есть консул :)

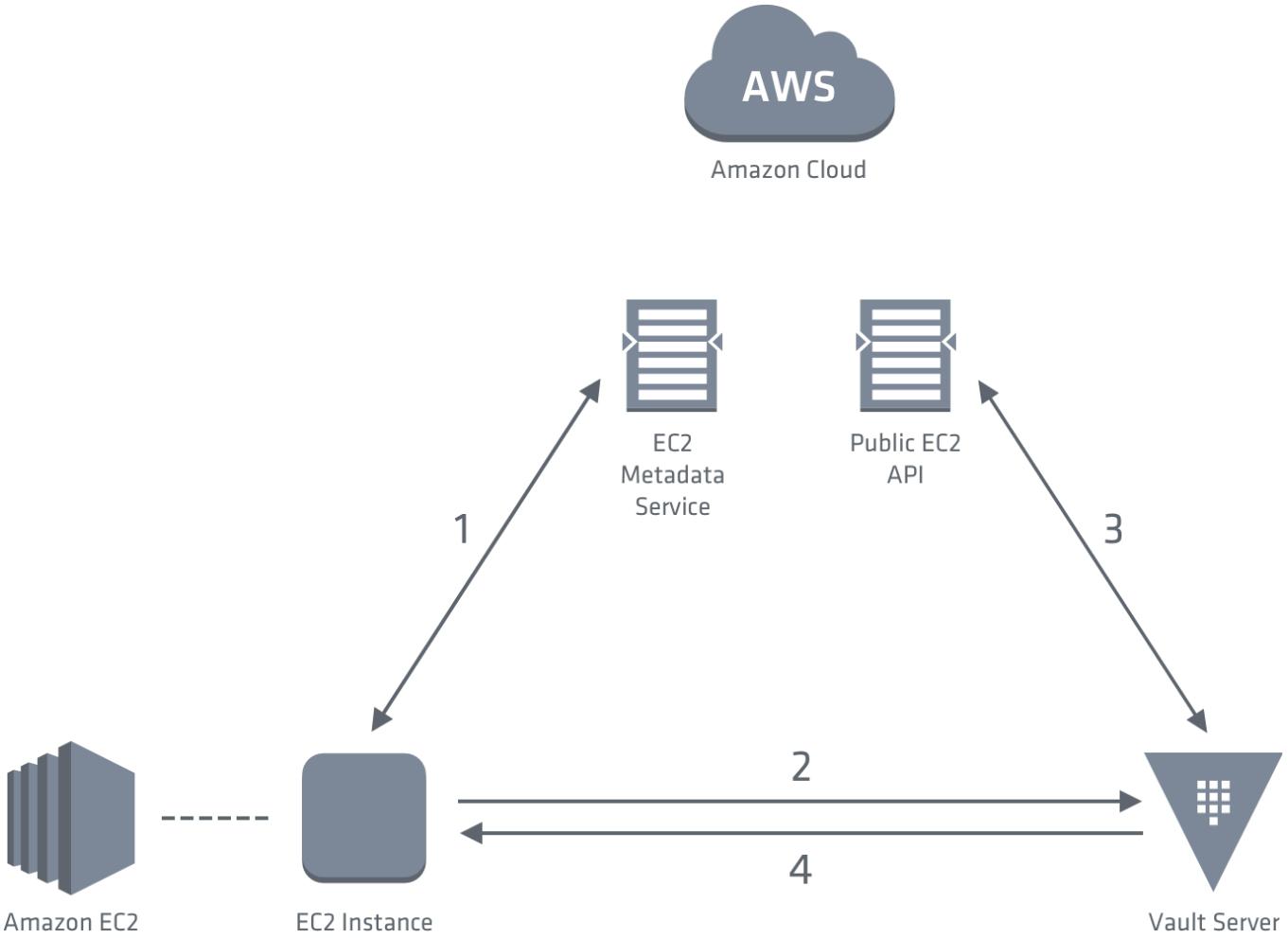
Управление секретами используя бэкенд

- AWS
- Consul
- Mongo
- Postgres
- Nomad
- RabbitMQ
- другие

Авторизация в vault

- AWS EC2
- AWS IAM
- Tokens
- Kubernetes
- LDAP
- другие

AWS EC2


AWS
Amazon Cloud
EC2
Metadata
Service
Public EC2
API
1
3
2
4
Amazon EC2
EC2 Instance
Vault Server

AWS EC2 - что происходит

- Получить подпись машины
- Запрос на токен с подписью отправляется в Vault
- Vault сверяет подпись машины через AWS API
- Vault token выдается в ответ на запрос
- Приложение использует токен для логина

Интересная часть

Kubernetes - что такое

- Надежно хранимые ресурсы
- Контроллеры
- Да и всё

Kubernetes - ServiceAccount

- Биндинг политик доступа k8s
- Опциональный маунт k8s API токена (jwe)
- Доступ к секретам k8s

Kubernetes - ServiceAccount - пример

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: myrestrictedaccount
  namespace: default
automountServiceAccountToken: true
```

Kubernetes - минимальный набор запуска

Приложения описываются ресурсом Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: app
spec:
  serviceAccountName: myrestrictedaccount
  containers:
    - name: app
      image: myapp:latest
```

Kubernetes + Vault

- Регистрируем кластер в Vault
- Ассоциируем ServiceAccount с ролью в Vault
- jwe API token k8s дает возможность выполнить логин

Terraform

- Infrastructure as code
- Инфраструктура как состояние
- Множество разных провайдеров

Terraform + Kubernetes - Vault policy

```
resource "vault_policy" "example" {
  name = "some_policy"
  policy = <<EOT
path "secret/example" {
  policy = "read"
}
EOT
}
```

Terraform + Kubernetes - Vault role

```
resource "vault_generic_secret" "vault_role" {
  path = "auth/kubernetes/role/vaultrolename"
  data_json = <<EOF
{
  "bound_service_account_names":      ["myrestrictedaccount"
  "bound_service_account_namespaces": ["default"],
  "policies": ["some_policy"],
  "ttl": 3600
}EOF
}
```

Kubernetes - SecretClaim

https://github.com/roboll/kube-vault-controller

```
kind: SecretClaim
apiVersion: vaultproject.io/v1
metadata:
  name: some-secret
spec:
  type: Opaque
  path: secret/example
  renew: 3600
```

Резюме (не то что для найма)

- Vault решает задачи безопасного хранения секретов
- Тулинг работы с Vault и его интеграцией достаточно зрел
- Интеграция с Kubernetes доступна на разных уровнях
- В очередной раз спасибо Hashicorp

Вопросы?